

LAPORAN PENELITIAN MANDIRI

ANALISIS DAN PERANCANGAN
DATAWAREHOUSE DAN INTELEGENSI BISNIS
UNTUK SUBJEK AKADEMIK
UNIVERSITAS ATMA JAYA YOGYAKARTA



Disusun oleh:

Irya Wisnubhadra, S.T.,M.T.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
2010

LEMBAR PENGESAHAN LAPORAN PENELITIAN

No proposal : _____

1.	a. Judul Penelitian	:	ANALISIS DAN PERANCANGAN DATAWAREHOUSE DAN INTELEGENSI BISNIS UNTUK SUBJEK AKADEMIK UNIVERSITAS ATMA JAYA YOGYAKARTA
	b. Macam Penelitian	:	Lapangan
2.	Peneliti		
	a. Nama	:	Irya Wisnubhadra, S.T., M.T.
	b. Jenis Kelamin	:	Laki-laki
	c. Usia saat pengajuan proposal	:	40 tahun 0 bulan
	d. Jabatan Akademik/Gol	:	Lektor / IIIc
	e. Fakultas / Program Studi	:	Teknologi Industri / Teknik Informatika
3.	Jumlah Peneliti	:	1 (satu) orang
4.	Personalia Peneliti	:	-
5.	Lokasi Penelitian	:	Yogyakarta
6.	Jangka Waktu Penelitian	:	6 (enam) bulan
7.	Biaya yang diajukan	:	3.950.000,- (Tiga juga sembilan ratus lima puluh ribu rupiah)

Ketua Program Studi Teknik Informatika

Yogyakarta, September 2010

Peneliti Utama

Prof. Ir. Suyoto, M.Sc., Ph.D.

Irya Wisnubhadra, S.T., M.T.

Dekan FTI UAJY

Ketua LPPM UAJY

Ir. B. Kristyanto M.Eng., Ph.D

Dr. Shellyana Junaedi, SE, M.Si

ANALISIS DAN PERANCANGAN DATA WAREHOUSE DAN INTELEGENSI BISNIS UNTUK SUBJEK AKADEMIK UNIVERSITAS ATMA JAYA YOGYAKARTA

Abstrak

Sistem Informasi pada era informasi saat ini telah menjadi kebutuhan dasar pada institusi untuk melakukan transaksi bisnis yang efisien dan efektif, termasuk juga didalamnya adalah institusi pendidikan. Universitas Atma Jaya Yogyakarta sebagai salah satu universitas terkemuka di Indonesia telah menggunakan Sistem Informasi pada berbagai aspek seperti pembelajaran, administrasi, penelitian dan kemahasiswaan, alumni dan kerjasama. Sistem Informasi tersebut ada pada masing-masing unit di universitas yaitu fakultas, biro, lembaga, pusat dan kantor.

Universitas Atma Jaya Yogyakarta telah mempunyai Sistem Informasi Akademik standar yang telah digunakan untuk keperluan pengelolaan akademik pada Fakultas dan unit-unit terkait pada Universitas. Sistem Informasi ini telah menghasilkan banyak data dari tahun ke tahun yang tampak seperti dokumen tidak berguna yang ada pada Server. Universitas sebagai lembaga pendidikan harus mempertanggung-jawabkan kegiatannya kepada para *stakeholder*. Tanggung jawab tersebut disajikan dalam bentuk laporan tahunan, portofolio fakultas dan universitas, borang akreditasi dan evaluasi diri. Laporan-laporan ini merupakan dokumen yang selalu diminta oleh regulator yang dapat juga dimanfaatkan untuk pengambilan keputusan strategis yang diharapkan mampu memperbaiki kinerja akademik, kinerja bisnis dari universitas.

Saat ini, laporan-laporan yang dihasilkan di Universitas Atma Jaya Yogyakarta selalu dibuat berdasarkan permintaan insidental. Universitas Atma Jaya Yogyakarta belum mempunyai aplikasi *reporting* yang terintegrasi yang dapat membantu pimpinan universitas untuk mengambil keputusan untuk menjawab berbagai persoalan seperti persoalan promosi, pengendalian sumber daya manusia, peningkatan mutu akademik, dan lain sebagainya.

Penelitian ini akan melakukan analisis dan perancangan *data warehouse* dan intelegensi bisnis, yaitu aplikasi yang dapat melakukan pelaporan yang terintegrasi, berorientasi subjek dari waktu ke waktu yang merupakan platform dasar untuk membantu pimpinan universitas untuk mengambil keputusan yang relevan untuk meningkatkan kinerja untuk bidang akademik. Terlebih dengan kebutuhan Penjaminan Mutu Akademik eksternal dan internal yang mengharuskan setiap universitas/fakultas/program studi melaporkan setiap kegiatan akademiknya.

BAB I PENDAHULUAN

I.1. Latar Belakang

Meningkatnya persaingan dalam dunia pendidikan tinggi menuntut pengelolaan yang berubah. Saat ini kecepatan dalam pembuatan keputusan baik yang bersifat internal maupun eksternal sangat diperlukan untuk meningkatkan daya saing sebuah perguruan tinggi. Untuk membantu dalam meningkatkan kecepatan maupun kualitas keputusan yang dibuat, maka adanya sebuah sistem informasi yang baik dan handal didukung dengan penggunaan teknologi informasi yang tepat akan sangat membantu dalam pembuatan keputusan.

Saat ini sebuah perguruan tinggi harus dapat memberikan layanan yang baik kepada seluruh *stakeholders* seperti kepada mahasiswa, alumni, orang tua mahasiswa, dunia industri, masyarakat dan pemerintah. Salah satu layanan yang dibutuhkan adalah layanan informasi akademik seperti nilai mahasiswa, hasil studi, proses studi, kewajiban pembayaran dan sebagainya. Untuk mendukung pemberian layanan tersebut maka dibutuhkan satu sistem informasi akademik yang handal dan baik.

Kebutuhan akan adanya sistem informasi akademik terintegrasi di tingkat universitas ini juga menjadi sasaran dari RENATA 2004/2005 – 2009/2010 yang menyatakan bahwa:

- a. Terciptanya sistem informasi akademik berbasis media elektronik yang terintegrasi.
- b. Memiliki sistem informasi yang terintegrasi di semua kampus.
- c. Memiliki manajemen sistem informasi yang baik, mulai dari perencanaan, pengelolaan teknologi informasi (*hardware, software* dan *brainware*) serta implementasi dan evaluasi.
- d. Pemanfaatan teknologi informasi untuk pengembangan bidang administrasi, sarana riset, akademik dan pengambilan keputusan.

Sistem Informasi dan Teknologi Informasi telah banyak dimanfaatkan pada semua lini dalam lingkungan Universitas Atma Jaya Yogyakarta. Sistem Informasi dan Teknologi Informasi digunakan untuk menunjang kegiatan Pembelajaran, kegiatan Penelitian, Pengabdian pada masyarakat dan kegiatan Administrasi untuk semua aspek kegiatan universitas dari akademik, kemahasiswaan, alumni, kerjasama, kepegawaian, keuangan, sumber daya manusia, penelitian, pengabdian masyarakat, pendampingan mahasiswa, pusat bahasa, pusat studi dll. Penggunaan Sistem Informasi dan Teknologi Informasi tersebut adalah :

Kegiatan Pembelajaran :

1. Situs web untuk keperluan pembelajaran : situs E-Learning misalnya <http://kuliah.inf.uajy.ac.id>, <http://mtf.inf.uajy.ac.id>, <http://course.fe.uajy.ac.id>, <http://kuliah.ind.uajy.ac.id>
2. Sistem Informasi Akademik (SIATMA) : Sistem Informasi Akademik pada tiap fakultas yang mengelola kegoi

Kegiatan Administrasi :

1. Email Server , <http://mail.uajy.ac.id> untuk civitas akademis Universitas Atma Jaya Yogyakarta
2. Sistem Informasi Pengelolaan Soal Penerimaan Mahasiswa Baru
3. Sistem Informasi Penerimaan Mahasiswa Baru.
4. Sistem Informasi Anggaran dan Akuntansi untuk keperluan pengelolaan cashflow, pencairan dana, laporan pertanggungjawaban serta pembukuan keuangan.

5. Sistem Informasi Billing Telepon.
6. Sistem Informasi Pembayaran Mahasiswa yang *real time* dengan bank-bank satu atap dengan Universitas Atma Jaya Yogyakarta.
7. Sistem Informasi Keuangan dan Penggajian.
8. Sistem Pendukung Keputusan Academic Budget Guidance System (ABGS)
9. Sistem Informasi Perpustakaan, yaitu NCI-Bookman, CDS-ISIS, Sistem Informasi Pintu Elektronik, Sistem Informasi Denda dan Box Service.
10. Sistem Informasi Manajemen Inventaris.
11. Expert System Penerimaan Dosen Baru.
12. Search Engine di web untuk pencarian Undang Undang yang ada di Universitas Atma Jaya Yogyakarta.

Kegiatan Kemahasiswaan, Alumni dan Kerjasama :

1. Situs web untuk lembaga kemahasiswaan, <http://students.inf.uajy.ac.id>
2. Situs alumni untuk berkomunikasi antar alumni, <http://alumni.inf.uajy.ac.id>, <http://www.uajy.net>, <http://www.kamajaya.org>,
3. Situs untuk kerjasama baik dengan industri dan lembaga pendidikan, <http://www.uajy.ac.id>,

Dari begitu banyak sistem informasi yang ada di Universitas Atma Jaya Yogyakarta sampai saat ini belum terintegrasi secara universitas atau masih bersifat departemental karena masing-masing dibangun pada waktu yang berbeda dan menggunakan platform yang berbeda pula. Banyak transaksi yang mengharuskan keterkaitan transaksi dari satu unit ke unit yang lain tidak dapat diproses dengan baik. Laporan-laporan baik akademik maupun non akademik yang harus dihimpun dari berbagai unit sulit untuk didapatkan secara cepat. Sifat departemental dari sistem informasi ini menyebabkan ketidakakuratan dan ketidakkonsistenan data yang menyebabkan informasi yang dihasilkan baik dalam bentuk laporan maupun data yang digunakan kembali dalam transaksi menjadi tidak valid. Ketidakvalidan data pada akhirnya juga berdampak pada pengambilan keputusan yang diambil pada manajemen puncak.

Pengambilan keputusan di bidang akademik, promosi, kehumasan dan sumber daya manusia oleh manajemen puncak membutuhkan data yang terintegrasi, tidak departemental, relevan untuk jangka waktu lama. Data dari masing-masing sistem informasi operasional/transaksional harus diintegrasikan, dikelompokkan berdasar subjek tertentu dan berdimensi waktu yang relevan sehingga Laporan/Report dapat dibangkitkan dengan cepat dan mudah dan tidak mengganggu sistem operasional yang pada akhirnya dapat digunakan membantu manajemen puncak universitas untuk mengambil keputusan strategis. Penelitian ini akan melakukan analisis dan perancangan data warehouse dan aplikasi intelegensi bisnis yaitu aplikasi untuk membantu manajemen puncak universitas. Hasil rancangan berupa basis data terpisah dari sistem informasi operasional yang menyediakan data dalam bentuk yang efisien bagi pelaporan yang diinginkan oleh manajemen puncak.

I.2. Perumusan Masalah

Rumusan masalah yang akan dijawab melalui penelitian yang akan dilakukan adalah sebagai berikut:

1. Bagaimana menganalisis dan merancang data warehouse dan aplikasi intelegensi bisnis bagi Universitas Atma Jaya Yogyakarta sehingga menghasilkan basis data dalam bentuk yang efisien bagi pelaporan yang diinginkan oleh manajemen puncak ?

I.3. Tujuan Penelitian:

Penelitian ini bertujuan:

- Menganalisis kebutuhan data warehouse dan aplikasi intelegensi bisnis untuk subjek akademik pada Universitas Atma Jaya Yogyakarta.
- Merancang data warehouse dan aplikasi intelegensi bisnis untuk subjek akademik Universitas Atma Jaya Yogyakarta.

I.4. Manfaat Hasil Penelitian:

Manfaat yang didapat dari penelitian ini adalah:

- Universitas Atma Jaya Yogyakarta mempunyai rancangan data warehouse dan aplikasi intelegensi bisnis yang dapat digunakan sebagai dasar pengambilan keputusan yang berhubungan dengan akademik.
- Universitas Atma Jaya Yogyakarta mempunyai rancangan infrastruktur data warehouse bidang akademik yang dapat dikembangkan untuk bidang lain.

I.5. Metodologi:

Penelitian ini dilakukan dengan melakukan sejumlah aktivitas yang berkaitan, antara lain:

- Studi Pustaka
Studi pustaka meliputi studi mengenai data warehouse dan aplikasi intelegensi bisnis untuk subjek akademik.
- Melakukan Analisis Kebutuhan Data Warehouse dan Aplikasi Intelegensi Bisnis.
Setelah informasi penelitian terkini dirangkum pada studi pustaka, langkah berikutnya adalah melakukan analisis dan menspesifikasikan kebutuhan data warehouse dan aplikasi intelegensi bisnis. Pada bagian dokumen analisis data warehouse dalam bentuk Information Package dibuat.
- Perancangan Data Warehouse dan Aplikasi Intelegensi Bisnis.

Setelah kebutuhan terdokumentasi, maka langkah selanjutnya adalah merancang Data Warehouse dan Aplikasi Intelegensi Bisnis. Perancangan di dokumentasikan dalam bentuk skema basis data pada deskripsi perancangan perangkat lunak (DPPL).

BAB III

TINJAUAN PUSTAKA

3.1 Intelengensi Bisnis

Pengertian Intelengensi Bisnis

Seorang analis dari Gartner Group, mendefinisikan intelegensi bisnis sebagai:

Sebuah proses dalam mengubah data menjadi informasi dan melalui penemuan-penemuan iteratif mengubah informasi tersebut menjadi bussiness intelligence (intelegensi bisnis). Kuncinya adalah bahwa intelegensi bisnis merupakan sebuah proses yang cross functional, sesuai dengan pemikiran manajemen pada saat ini, dan tidak dihadirkan dalam istilah TI (Teknologi Informasi)(Dresner,2002).

Definisi lain dari intelegensi bisnis:

Business Intelligence menjelaskan tentang suatu konsep dan metode bagaimana untuk meningkatkan kualitas pengambilan keputusan bisnis berdasarkan sistem yang berbasiskan data. BI seringkali dipersamakan sebagaimana briefing books, report and query tools, dan sistem informasi eksekutif. BI merupakan sistem pendukung pengambilan keputusan yang berbasiskan data-data (Power,2002).

Jadi secara singkat dapat disimpulkan intelegensi bisnis adalah sebuah aplikasi atau teknologi yang memiliki kemampuan untuk memahami pola antar data yang dimiliki dalam konteks tertentu sehingga dapat digunakan dalam mendukung pengambilan keputusan dalam bidang bisnis.

Latar Belakang Intelengensi Bisnis

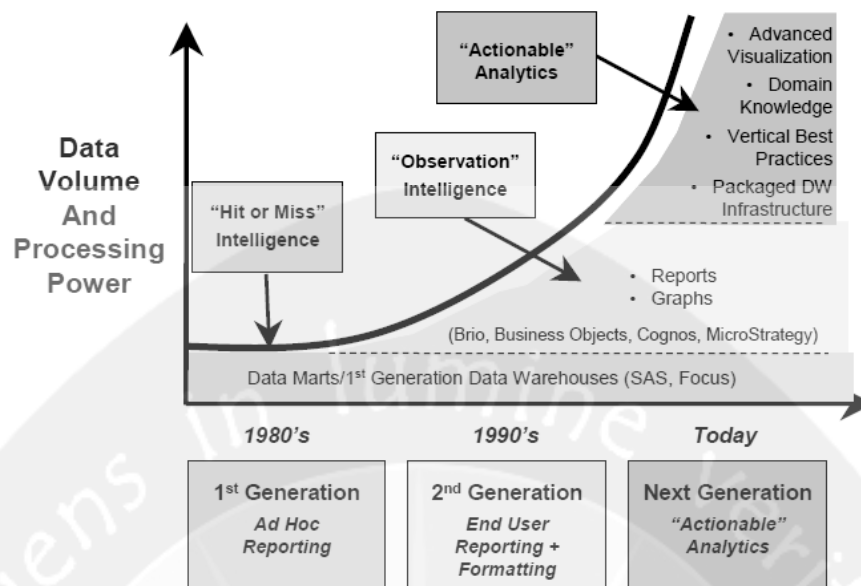
Pengambilan keputusan bergaya *entrepreneur* yang cenderung mengandalkan intuisi menjadi kurang relevan di tengah lingkungan persaingan yang semakin kompleks. Naik turunnya penjualan sebuah produk tidak lagi hanya ditentukan oleh murah dan mahal nya harga jual atau baik buruknya kualitas produk. Ternyata banyak faktor dominan lain seperti kondisi demografi, geografi, penawaran pesaing, dan bahkan faktor emosional. Perusahaan memerlukan alat bantu yang mampu mengolah data-data yang dimilikinya menjadi informasi bernilai lebih dan tidak hanya terpaku pada angka-angka mati. Alat bantu tersebut harus mampu menterjemahkan berbagai hal implisit dan gejala penting dari sekumpulan data. Salah satu jawabannya adalah intelegensi bisnis.

Intelegensi bisnis merupakan sistem dasar bagi hampir seluruh kondisi yang melibatkan pembuatan keputusan bisnis dan formulasi strategi. Memang intelegensi bisnis tak secara langsung menyelesaikan berbagai tantangan, namun intelegensi bisnis mampu dan andal dalam mengidentifikasi suatu solusi dan langkah maju. Caranya dengan menyediakan informasi yang relevan dan mudah digabungkan dengan pengambilan keputusan dan proses formulasi strategi. Hal inilah yang tidak mampu dilakukan oleh sistem komputer lainnya. Fakta bahwa bisnis terus berubah, di sinilah intelegensi bisnis mengambil peran agar bisnis luwes beradaptasi terhadap segala perubahan.

Berikut ini adalah contoh-contoh masalah yang dapat diperoleh jawabannya menggunakan intelegensi bisnis:

1. Manager Promosi ingin menganalisis pengaruh tiap jenis media iklan di koran, majalah, dan TV terhadap penjualan produk.
2. Manager HRD dapat menganalisis pengaruh kenaikan gaji terhadap peningkatan produktivitas pekerja di lantai pabrik.
3. Manajer Penjualan ingin mengetahui pengaruh musim dan kepadatan penduduk terhadap penjualan es krim di tiap daerah.

3.2.1 Perkembangan Intelegensi Bisnis



Gambar 3.1. Perkembangan Intelegensi Bisnis (Asif,2010)

Perkembangan intelegensi bisnis dibagi ke dalam tiga generasi. Ketiga generasi tersebut adalah *1st Generation*, *2nd Generation* dan *Next Generation* (Asif,2010). Pada *first generation*, intelegensi bisnis mampu mengeksekusi *query* yang spesifik dan *customized* untuk menunjukkan data tertentu, misal data mengenai penjualan bulan April. Seiring perkembangan kemampuan *processing* dan volume data, intelegensi bisnis pada tahap *second generation* menjadi mampu untuk menghasilkan laporan dalam bentuk tabel-tabel dan grafik sehingga mempermudah tindakan observasi. Pada tahap *next generation*, intelegensi bisnis dituntut untuk mampu mengandung basis pengetahuan (*knowledge base*) dari praktisi bisnis sehingga dapat memberikan saran kepada pengguna tentang tindakan yang harus diambil untuk mengantisipasi perilaku pasar.

Sebuah penelitian mengenai perkembangan intelegensi bisnis mengatakan bahwa 20% dari organisasi yang berdiri pada tahun 2010 akan memiliki *industry-specific analytic application* untuk mendukung sistem intelegensi bisnis yang mereka miliki. Selain itu, dikatakan pula bahwa pada tahun 2012, 40% dari total budget sebuah unit bisnis akan terserap untuk pengelolaan intelegensi bisnis yang dimiliki (anonim,2010).

3.2.2 Manfaat Intelegensi Bisnis bagi Perusahaan

Stevan (2008) menyebutkan manfaat intelegensi bisnis yang akan memberikan nilai bagi perusahaan:

1. Keuntungan

Perusahaan yang telah mengimplementasikan intelegensi bisnis akan mampu menuai keuntungan keuangan dari implementasi tersebut. Dengan struktur implementasi intelegensi bisnis yang baik, perusahaan akan cerdas mengadaptasikan tingkat layanan yang ditawarkan demi menjawab kebutuhan pelanggan.

2. Pangsa pasar

Salah satu cara efektif menggenjot keuntungan dan stabilitas pasar adalah dengan melebarkan pangsa pasar perusahaan. Pebisnis baru yang terus bermunculan selalu berusaha menggerogoti pangsa pasar perusahaan yang telah mapan. Perusahaan dengan keuangan yang lebih stabil tentu akan mampu bertahan, sisanya bisa jadi akan terlibas persaingan. Perusahaan yang mengimplementasikan intelegensi bisnis, peluangnya akan lebih besar untuk tetap bertahan jika mereka mampu bereaksi dengan cepat dan dengan keputusan yang lebih baik mempertahankan pangsa pasar.

3. Pengambilan keputusan

Keputusan yang dibuat oleh perusahaan harus yang terbaik dari pilihan yang ada. Dengan menerapkan intelegensi bisnis, para petinggi perusahaan dapat mengambil keputusan terbaik berdasarkan informasi-informasi yang diperoleh dari intelegensi bisnis.

3.2.3 Faktor-Faktor yang Mengakibatkan Kegagalan Intelegensi Bisnis

Mantfled (2006) menyebutkan terdapat beberapa faktor yang harus dihindari agar implementasi intelegensi bisnis di suatu organisasi berjalan sukses. Hal ini sangat penting diperhatikan karena upaya imlementasi intelegensi bisnis biasanya akan membutuhkan sumber daya (dana, waktu, tenaga) yang relatif cukup besar. Faktor-faktor tersebut adalah :

1. Perencanaan yang kurang matang

Implementasi intelegensi bisnis tidak mungkin berhasil tanpa perencanaan yang matang. Kondisi tersebut antara lain ditunjukkan dengan adanya rendahnya konsistensi dukungan pimpinan terhadap proyek intelegensi bisnis itu sendiri dan rendahnya tingkat kerjasama antarbagian di organisasi dalam upaya mewujudkan intelegensi bisnis. Selain itu, kurang jelasnya kebutuhan informasi yang ingin didapatkan dari pengembangan intelegensi bisnis juga berpotensi menurunkan tingkat keberhasilan.

2. Kualitas data yang tidak/kurang baik

Intelegensi bisnis tidak akan dapat digunakan dengan baik jika data yang akan dianalisis merupakan data yang tidak/kurang baik kualitasnya. Data yang kurang baik akan menghasilkan informasi yang kurang baik dalam pengambilan keputusan (*garbage in = garbage out concept*).

3. Kurangnya mengantisipasi terhadap perubahan di organisasi

Sistem intelegensi bisnis beserta implementasinya seringkali mengalami perubahan kebutuhan dan organisasi patut mengantisipasi hal tersebut. Perubahan yang terjadi di organisasi pun juga membutuhkan antisipasi pada sistem intelegensi bisnis yang dipakai.

4. Pengadaan sistem intelegensi bisnis yang *one-stop shopping*

Sampai dengan saat ini, belum ada sistem intelegensi bisnis yang siap pakai untuk semua jenis organisasi yang membutuhkannya. Untuk itu pengadaan intelegensi bisnis di suatu organisasi memerlukan suatu proses penyempurnaan yang berkelanjutan dan bukan hanya sekadar pembelian sistem yang sekali beli dapat digunakan seterusnya tanpa penyempurnaan.

5. Pengembangan intelegensi bisnis hanya mengandalkan tenaga *outsourcing*

Pihak *outsourcing* (*vendor*) tidak mungkin mengetahui informasi yang lengkap dan detail mengenai kebutuhan organisasi tersebut dengan sendirinya tanpa bantuan dari pegawai organisasi yang bersangkutan. Pihak *outsourcing* hanya bertindak sebagai tenaga yang membantu membuat sistem, tapi bentuk sistem dan kebutuhan apa saja yang diperlukan hanya organisasi lah yang mengetahui dengan baik.

3.3 OLTP (*Online Transactional Processing*) dan OLAP (*Online Analytical Processing*)

OLTP dan OLAP adalah model-model dari basis data, yang mempunyai tujuan dan kegunaan berbeda. OLTP adalah bentuk dari sistem informasi tradisional yang lebih dahulu dikenal untuk membantu operasi-operasi harian, sedangkan OLAP muncul karena dirasakannya kebutuhan akan sebuah sistem informasi yang berorientasi untuk membantu pengambilan keputusan.

3.3.1 OLTP (*Online Transactional Processing*)

Berikut ini merupakan salah satu definisi OLTP:

OLTP (Online Transactional Processing) adalah sistem yang memproses suatu transaksi secara langsung melalui komputer yang terhubung dalam jaringan (Febrian, 2004).

OLTP ditujukan untuk mendukung proses-proses transaksi harian dari sebuah organisasi. Sistem ini sangat berguna untuk membuat roda bisnis terus berputar karena digunakan untuk menangani proses-proses operasional dari perusahaan. Tipikal dari OLTP ini adalah untuk menangani sejumlah besar transaksi, yang dilakukan oleh sejumlah besar pengguna secara simultan dengan cepat (*real time*). Beberapa aplikasi OLTP antara lain: *electronic banking, order processing, employee time clock systems, dan e-commerce*. OLTP mempunyai karakteristik yaitu transaksi hanya mengakses sebagian kecil basis data, pemutakhiran relatif sering dilakukan, serta transaksi singkat dan sederhana.

3.3.2 OLAP (*Online Analytical Processing*)

Berikut ini adalah sebuah definisi dari OLAP:

Online Analytical Processing adalah sintesis dinamik, analisis dan konsolidasi dari data multidimensional yang sangat besar (Codd et all, 1993).

OLAP memanipulasi dan menganalisis data bervolume besar dari berbagai perspektif (multidimensi). Oleh karena itu OLAP seringkali disebut analisis data multidimensi. OLAP bekerja dengan data dalam bentuk multidimensi, umumnya bentuk tiga dimensi diwujudkan ke dalam bentuk kubus data. Tujuan OLAP adalah menggunakan informasi dalam sebuah basis data (*data warehouse*) untuk memandu keputusan-keputusan yang strategis. Karakteristik OLAP yaitu permintaan data sangat kompleks, jarang ada pemutakhiran, transaksi mengakses banyak bagian dalam basis data.

Beberapa contoh permintaan yang ditangani oleh OLAP: Berapa jumlah penjualan dalam kuartal pertama? Berapa jumlah penjualan per kuartal untuk masing-masing kota? Tampilkan 5 produk dengan total penjualan tertinggi pada kuartal pertama. Kadangkala permintaan yang ditangani OLAP bisa diselesaikan dengan pernyataan SQL sederhana, tetapi dalam banyak kasus tidak dapat diekspresikan dengan SQL.

OLAP dapat digunakan untuk melakukan konsolidasi, *drill-down*, dan *slicing and dicing*.

1. Konsolidasi melibatkan pengelompokan data. Sebagai contoh kantor-kantor cabang dapat dikelompokkan menurut kota atau bahkan propinsi. Transaksi penjualan dapat ditinjau menurut tahun, triwulan, bulan, dan sebagainya. Kadangkala istilah *rollup* digunakan untuk menyatakan konsolidasi.
2. *Drill-down* adalah suatu bentuk yang merupakan kebalikan dari konsolidasi, yang memungkinkan data yang ringkas dijabarkan menjadi data yang lebih detail. Sebagai contoh, mula-mula data yang tersaji didasarkan pada kuartal pertama. Jika dikehendaki, data masing-masing bulan pada kuartal pertama tersebut bisa diperoleh, sehingga akan tersaji data bulan Januari, Februari, Maret, dan April.
3. *Slicing and dicing* (atau dikenal dengan istilah *pivoting*) menjabarkan pada kemampuan untuk melihat data dari berbagai sudut pandang. Data dapat diiris-iris atau dipotong-potong berdasarkan kebutuhan. Sebagai contoh, dapat diperoleh data penjualan berdasarkan semua lokasi atau hanya pada lokasi-lokasi tertentu.

Berdasarkan struktur basis datanya OLAP dibedakan menjadi 3 kategori utama :

1. *Multidimensional Online Analytical Processing* (MOLAP)
Sistem OLAP pada masa awal menggunakan larik multidimensi di dalam memori untuk menyimpan data kubus.
2. *Relational Online Analytical Processing* (ROLAP)
Format pengolahan OLAP yang melakukan analisis data secara dinamis yang disimpan dalam basis data relasional bukan pada basis data multidimensi.
3. *Hybrid Online Analytical Processing* (HOLAP)
Merupakan kombinasi antara ROLAP dengan MOLAP. HOLAP dikembangkan untuk mengkombinasikan antara kapasitas data pada ROLAP yang besar dengan kemampuan proses pada MOLAP.

3.3.3 Perbedaan OLTP dan OLAP

OLTP merupakan tugas utama dari relasi DBMS tradisional, yaitu mendukung operasi harian seperti: *purchasing, inventory, banking, manufacturing, payroll, registration, accounting*, dsb. Sedangkan OLAP merupakan tugas utama dari sistem *data warehouse*, yaitu analisa data dan pengambilan keputusan.

Perbedaan-perbedaan antara OLTP dan OLAP ditunjukkan oleh tabel berikut:

Tabel 3.1. Perbedaan OLTP dan OLAP (Anonim,2009)

Kategori	OLTP	OLAP
Tujuan	Mengotomatisasi bisnis	Mengoptimalkan bisnis
Penggunaan	Proses transaksi	Pelaporan, analisis, perencanaan, peramalan
Tipe pengguna	clerk, profesional IT	Manager, analis, eksekutif
Skema DB	2 dimensi, ternormalisasi	Multidimensi, denormalisasi
Desain DB	RDBMS, Berorientasi aplikasi	ROLAP, MOLAP, HOLAP (star, snow-flake), Berorientasi subjek
Operasi	insert, update, delete	select
Frekuensi refresh	Segera	Periodic (tahunan, bulanan)
Frekuensi penggunaan	berulang	Khusus untuk sesuatu maksud
Level data	Detail, data transaksi	Ringkasan data
Penggunaan data	Data terkini	Data historis (lampau)
Sumber data	Hanya data pada aplikasi tertentu	Data terintegrasi dari berbagai macam aplikasi
unit kerja	Transaksi pendek, sederhana	Query kompleks
record yg diakses	puluhan	jutaan

Basis data OLTP dirancang untuk kemudahan data *entry* dan bukan untuk keperluan *report*. Membuat *report* dari basis data OLTP dengan struktur yang kompleks akan sangat sulit dilakukan. Selain itu proses pengambilan data pada saat *report* ditampilkan akan mempengaruhi performa basis data OLTP karena untuk analisis data yang diambil merupakan agregasi yang akan menghabiskan sumber daya *server*. Dengan basis data OLAP, data yang disimpan sudah berupa hasil agregasi yang akan mempercepat waktu dan performa basis data. Selain itu struktur data pada basis data OLAP juga akan memudahkan proses pembuatan *report* dan analisis data.

3.4 Data Warehouse

Dalam subbab ini akan dijelaskan mengenai pengertian *data warehouse*, latar belakang *data warehouse*, karakteristik *data warehouse*, dan komponen *data warehouse*.

3.4.1 Pengertian Data Warehouse

Menurut pelopor konsep dan istilah *data warehouse*, William Inmon, definisi dari *data warehouse* adalah:

Sebuah koleksi data yang berorientasi subjek, terintegrasi, non-volatile, dan time-variant dalam rangka mendukung keputusan-keputusan manajemen (Inmon, 1996).

Beberapa definisi lain dari beberapa sumber-sumber resmi sebagai berikut:

Data warehouse merupakan database yang bersifat analisis dan read only yang digunakan sebagai fondasi dari sistem penunjang keputusan (Poe, 1998).

Data warehousing merupakan basisdata relasional yang didesain lebih kepada query dan analisa dari pada proses transaksi, biasanya mengandung history data dari proses transaksi dan bisa juga data dari sumber lainnya. Data warehousing memisahkan beban kerja analisis dari beban kerja transaksi dan memungkinkan

organisasi menggabung/konsolidasi data dari berbagai macam sumber (Lane,2003).

Dari definisi-definisi yang dijelaskan tadi, dapat disimpulkan *data warehouse* adalah basis data yang saling bereaksi yang dapat digunakan untuk *query* dan analisis, bersifat orientasi subjek, terintegrasi, time variant, tidak berubah yang digunakan untuk membantu para pengambil keputusan.

3.4.2 Latar Belakang Data Warehouse

Sebelum *data warehouse* diperkenalkan di dunia bisnis, perusahaan mengalami kesulitan dalam memproses informasi yang dihasilkan menjadi suatu pengetahuan. Suatu hal yang sulit untuk menginterpretasikan informasi yang sedemikian banyak jumlahnya menjadi sebuah pengetahuan yang relevan dengan kepentingan perusahaan. Disinilah, fungsi utama dari sebuah *data warehouse*, karena dengan pengetahuan yang baik, manajemen akan dengan mudah mengambil keputusan-keputusan strategis berdasarkan data yang akurat dan informasi yang berkualitas.

Filosofi penting yang harus diingat dalam mengimplementasikan *data warehouse* adalah “*garbage in, garbage out*”, maksudnya adalah secanggih apapun *data warehouse* yang dimiliki jika data yang dimasukkan sudah salah, maka informasi yang dihasilkan pun menjadi salah. Mengambil keputusan berdasarkan informasi yang salah dapat menjadi bumerang bagi perusahaan.

3.4.3 Karakteristik Data Warehouse

Poniah (2001) menyebutkan karakteristik dari *data warehouse* adalah sebagai berikut:

1. Berorientasi Subjek

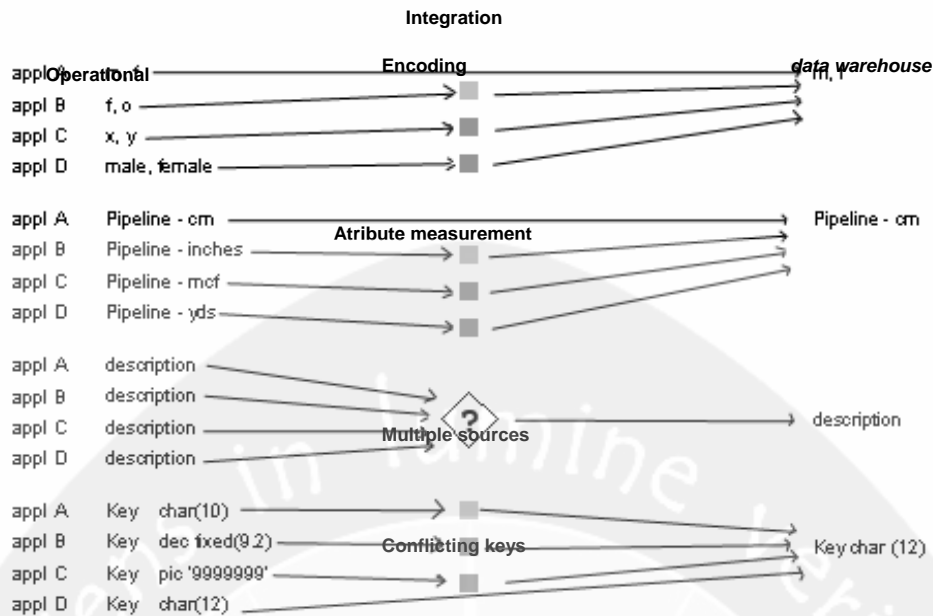
Data warehouse didesain untuk menganalisa data berdasarkan subjek-subjek tertentu dalam organisasi, bukan pada proses atau fungsi aplikasi tertentu. *Data warehouse* diorganisasikan disekitar subjek-subjek utama dari perusahaan (konsumen, produk dan penjualan) dan tidak diorganisasikan pada area-area aplikasi utama (*customer invoicing*, *stock control* dan *product sales*). Hal ini dikarenakan kebutuhan dari *data warehouse* untuk menyimpan data-data yang bersifat sebagai penunjang suatu keputusan, dari pada aplikasi yang berorientasi terhadap data.



Gambar 3.2. Perbedaan Data Warehouse dan Basis Data Operasional (Poniah,2001)

2. Terintegrasi

Sumber data yang ada dalam *data warehouse* tidak hanya berasal dari data operasional (*internal source*) tetapi juga berasal dari data di luar sistem (*external source*). Setiap perancang aplikasi memiliki kendali bebas untuk membuat keputusan perancangannya sendiri. *Data Warehouse* dapat menyimpan data-data yang berasal dari sumber-sumber yang terpisah kedalam suatu format yang konsisten dan saling terintegrasi satu dengan lainnya. Syarat integrasi sumber data dapat dipenuhi dengan berbagai cara seperti konsisten dalam penamaan variabel, konsisten dalam ukuran variabel, konsisten dalam struktur pengkodean, dan konsisten dalam atribut fisik dari data.



Gambar 3.3. Masalah Integrasi (Inmon, 1996)

3. Time-variant

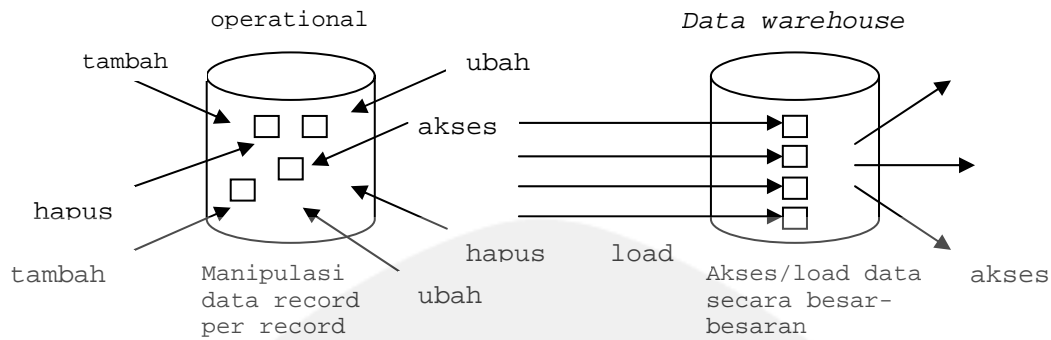
Sistem operasional mengandung data yang bernilai sekarang sedangkan data dalam *data warehouse* mengandung data tidak hanya data terkini tetapi juga data masa lampau yang akan digunakan dalam analisis dan pengambilan keputusan. Waktu adalah dimensi penting yang harus didukung oleh semua *data warehouse*. Data untuk analisis dari berbagai sumber berisi berbagai nilai waktu, misalkan harian, mingguan, dan bulanan.

Time variancy data pada *data warehouse* muncul dalam beberapa cara :

- Waktu untuk *data warehouse* lebih panjang daripada waktu pada sistem operasional. 60-90 hari adalah waktu yang normal untuk sistem operasi; 5-10 tahun adalah waktu yang normal untuk data pada *data warehouse*.
- Database* operasional terdiri dari “*current value*” data yang dapat di-*update*. Data pada *data warehouse* merupakan rangkaian *snapshots* yang rumit dan sulit yang diambil pada suatu waktu.
- Key structure* dari data operasional mungkin terdiri dari beberapa elemen waktu, seperti tahun, bulan, hari, dan sebagainya. *Key structure* dari *data warehouse* selalu terdiri dari beberapa elemen waktu.

4. Non-volatile

Data dalam basis data operasional akan secara berkala atau periodik dipindahkan ke dalam *data warehouse* sesuai dengan jadwal yang sudah ditentukan. Misal perhari, perminggu, perbulan, dan lain sebagainya. Data yang baru selalu ditambahkan sebagai suplemen bagi basis data itu sendiri dari pada sebagai sebuah perubahan. Basis data tersebut secara kontinyu menyerap data baru ini, kemudian secara inkrement disatukan dengan data sebelumnya. Basis data operasional bisa dibaca, diperbarui, dan dihapus. Tetapi pada *data warehouse* hanya ada dua kegiatan memanipulasi data yaitu *loading* data (mengambil data) dan akses data (mengakses *data warehouse* seperti melakukan *query* atau menampilkan laporan yang dibutuhkan, tidak ada kegiatan *updating* data).



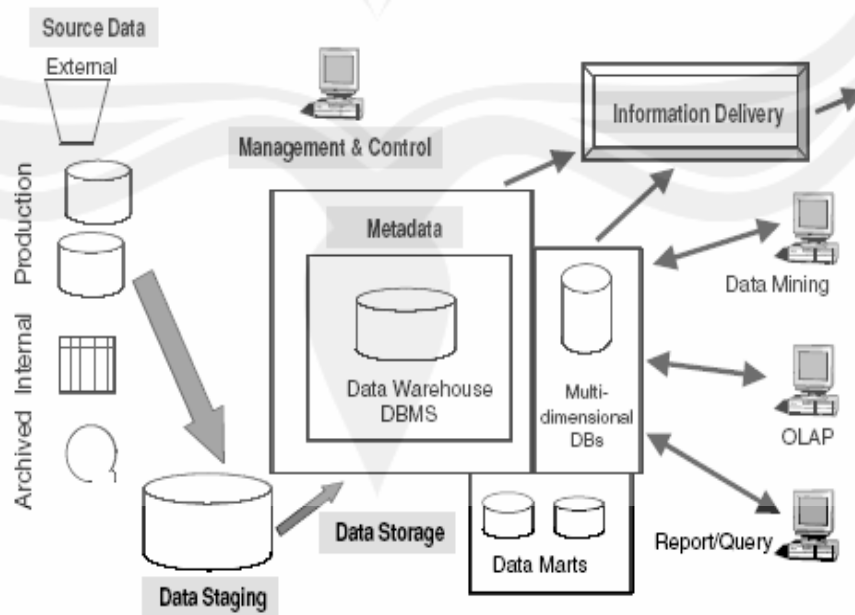
Gambar 3.4. Masalah Nonvolatility (Inmon, 1996)

5. Granularity

Pada sistem operasional data dibuat secara real-time sehingga untuk mendapatkan informasi langsung dilakukan proses query. Pada data warehouse pada menganalisis harus memperhatikan detail per level misalkan perhari, ringkasan perbulan, ringkasan per-tiga-bulan. Granularitas menunjuk pada level perincian atau peringkasan yang ada pada unit-unit data dalam *data warehouse*. Semakin banyak detail yang ada, maka semakin rendah level granularitas. Semakin sedikit detail yang ada, maka semakin tinggi level granularitas. Semakin tinggi level granularitas maka *query* yang dapat ditangani oleh *data warehouse* semakin terbatas. Semakin rendah level granularitas maka *query* yang dapat ditangani oleh *data warehouse* semakin banyak dan jawaban *query* yang diperolehpun semakin detail.

3.4.4 Komponen Data Warehouse

Pada subbab ini, akan dibahas tentang komponen-komponen *data warehouse*. Komponen *source data* berada di sebelah kiri. Komponen *data staging* sebagai blok pembangun berikutnya. Di tengah, dapat dilihat komponen *data storage* yang mengelola *data warehouse*, komponen ini tidak hanya menyimpan dan mengelola data, tetapi juga menjaga bagian data yang disebut *metadata repository*. Komponen *information delivery* berada di sebelah kanan, yang terdiri dari semua hal menyediakan informasi dari *data warehouse* bagi pengguna.



Gambar 3.5. Komponen Data Warehouse (Ponniah, 2001)

3.4.4.1 Komponen Source Data

Sumber data yang digunakan dalam *data warehouse* dapat dikelompokkan menjadi empat, yaitu:

1. *Production Data*

Data dalam kategori ini berasal dari berbagai sistem operasional dalam perusahaan. Berdasarkan kebutuhan informasi di dalam *data warehouse*, segmen-segmen data dipilih dari sistem operasional yang berbeda. Dalam proses ini, data-data yang ditangani kemungkinan besar berada dalam format yang bermacam-macam, kemungkinan juga berasal dari platform yang berbeda-beda. Lebih lanjut lagi, data-data tersebut didukung oleh sistem basis data dan sistem operasi yang berbeda-beda.

Karakteristik *production data* yang paling signifikan dan mengganggu adalah perbedaan (*disparity*). Tantangan terbesar adalah untuk membuat standarisasi dan mentransformasi data-data yang berlainan dari berbagai sistem operasional yang berbeda, mengkonversi data, dan mengintegrasikannya menjadi beberapa bagian yang kemudian akan disimpan dalam *data warehouse*.

2. *Internal Data*

Dalam setiap perusahaan, masing-masing *user* menjaga dokumen-dokumen, *spreadsheets*, profil-profil pelanggan mereka, terkadang bahkan basis data-basis data departemental. Inilah yang disebut dengan *internal data* atau data internal, bagian yang dapat menjadi berguna dalam sebuah *data warehouse*.

Jika sebuah perusahaan menjalankan bisnis dengan *one to-one basis* dan kontribusi dari tiap-tiap pelanggan di garis bawah adalah signifikan, maka profil-profil mendetail dari pelanggan dengan demografis yang luas merupakan hal penting dalam *data warehouse*. Meskipun banyak dari data ini mungkin didapatkan dari sistem produksi, sebagian besarnya dipegang oleh masing-masing karyawan dan departemen dalam *file-file* pribadi mereka.

Data-data internal dalam perusahaan seperti ini tidak dapat diabaikan. Data ini menambah kompleksitas dalam proses transformasi dan integrasi data sebelum dapat disimpan ke dalam *data warehouse*. Harus ditentukan strategi-strategi untuk mengambil data dari kertas-kertas kerja (*spreadsheets*), menemukan cara untuk mengambil data dari dokumen-dokumen tekstual, dan menghubungkannya ke dalam basis data-basis data departemental untuk mengumpulkan data yang saling berhubungan dari sumber-sumber tersebut.

3. *Archived Data*

Sistem operasional utamanya ditujukan untuk menjalankan suatu bisnis. Di setiap sistem operasional, kita secara berkala mengambil data lama dan menyimpannya di dalam *file-file* arsip.

Seperti telah disebutkan sebelumnya, sebuah *data warehouse* menyimpan *snapshot-snapshot* data historikal. Kita pada dasarnya memerlukan data-data historikal untuk melakukan analisis terhadap waktu. Untuk mendapatkan informasi historikal, kita melihat ke dalam data set-data set yang telah diarsipkan. Berdasarkan pada *requirements* dari *data warehouse* itu sendiri, kita harus memasukkan data-data historis yang memadai. Data tipe ini berguna untuk memahami pola-pola dan menganalisis tren-tren.

4. *External Data*

Banyak para eksekutif menggunakan data yang berasal dari sumber-sumber eksternal untuk meningkatkan persentase informasi yang mereka gunakan. Mereka menggunakan statistik-statistik yang berhubungan dengan industri mereka, yang dihasilkan oleh agensi-agensi eksternal. Mereka menggunakan data-data *market share* dari para kompetitor, serta nilai-nilai standar dari indikator-indikator keuangan terhadap bisnis mereka untuk mengetahui *performance* perusahaan.

Data eksternal dapat memberikan gambaran mengenai apa yang akan dilakukan atau apa yang telah dilakukan oleh suatu perusahaan. Data eksternal ini sangat dibutuhkan apabila suatu perusahaan ingin membandingkan perusahaannya dengan organisasi lain. Hal seperti inilah yang dapat meningkatkan kinerja perusahaan.

3.4.4.2 Komponen Data Staging

Setelah mengekstraksi data dari berbagai macam sistem operasional, data harus disiapkan untuk disimpan ke dalam *data warehouse*. Data-data hasil ekstraksi yang berasal dari beberapa sumber berlainan harus diubah, dikonversi, dan membuatnya siap dalam satu format yang sesuai untuk disimpan dan digunakan bagi keperluan *query* dan analisis. Tahap pembersihan ini dikenal juga dengan istilah ETL (*Extraction, Transformation, and Loading*).

Tahap pembersihan ini berlangsung di sebuah *staging area*. *Data staging* menyediakan sebuah tempat dengan satu set fungsi untuk membersihkan, mengubah, menggabungkan, mengkonversi, mencegah duplikasi data, dan menyiapkan data sumber untuk penyimpanan dan penggunaan dalam *data warehouse*.

Mengapa kita memerlukan sebuah tempat atau komponen terpisah untuk melakukan persiapan data? Dalam sebuah *data warehouse* kita menarik data dari berbagai sumber sistem operasional. Perlu diingat bahwa data-data dalam sebuah *data warehouse* adalah *subject oriented* dan melintasi aplikasi-aplikasi operasional. Oleh karena itu, sebuah *staging area* yang terpisah diperlukan untuk menyiapkan data untuk *data warehouse*.

1. Data Extraction

Fungsi ini berhubungan dengan sumber-sumber data yang banyak. Karena itu diperlukan penerapan teknik yang tepat untuk tiap-tiap data sumber. Data sumber mungkin berasal dari mesin-mesin yang berbeda dan dalam format-format data yang berlainan. Sebagian dari sumber data mungkin berada dalam sistem-sistem basis data relasional. Beberapa data mungkin berada dalam *legacy network* dan model-model data hirarkial. Banyak data-data mungkin masih berada dalam bentuk *flat files*. Bisa juga data yang diinginkan berasal dari *spreadsheets* dan data set-data set departemental lokal. Jadi, ekstraksi data bisa menjadi benar-benar kompleks.

Software-software untuk ekstraksi data sudah tersedia di pasaran. Untuk beberapa sumber data dapat digunakan software dari luar yang sesuai. Untuk sumber-sumber data lainnya, bisa jadi digunakan program-program yang dikembangkan sendiri. Membeli *software* dari luar mungkin memerlukan biaya yang besar. Program-program *in-house*, sebaliknya, bisa berarti biaya yang berkelanjutan untuk pembangunan dan perawatan.

Setelah proses pengestrakan data, dimanakah data disimpan untuk persiapan selanjutnya? Kita bisa menjalankan fungsi-fungsi ekstraksi dalam platform asal itu sendiri jika pendekatan tersebut sesuai untuk *framework* yang ada. Lebih sering lagi, tim pengembang *data warehouse* mengekstrak sumber ke dalam lingkungan fisik yang terpisah, yang membuat pemindahan data tersebut ke dalam *data warehouse* menjadi lebih mudah. Pada lingkungan yang terpisah, sumber data dapat diekstraksikan ke dalam sebuah grup *flat file*, atau sebuah basis data relasional *data-staging*, atau kombinasi dari keduanya.

2. Data Transformation

Setelah melalui tahap ekstraksi, data tersebut masih merupakan data mentah dan tidak dapat diaplikasikan ke *data warehouse*. Pertama-tama semua data hasil ekstraksi tersebut harus dibuat berguna dalam *data warehouse*. Mengolah informasi agar dapat digunakan (*useable*) untuk membuat keputusan strategis adalah prinsip utama dari *data warehouse*. Karena data-data operasional didapatkan dari banyak sistem lama, kualitas data pada sistem-sistem tersebut menjadi kurang baik untuk *data warehouse*. Kualitas data harus diperkaya dan dikembangkan sebelum dapat digunakan dalam *data warehouse*.

Sebelum memindahkan data yang telah diekstrak dari *source systems* ke dalam *data warehouse*, diperlukan beberapa macam transformasi data. Data harus ditransformasikan menurut standar-standar yang telah ditetapkan, karena data-data tersebut berasal dari *source systems* yang tidak sama. Harus dipastikan bahwa setelah data-data disatukan, kombinasi data tidak akan melanggar aturan-aturan bisnis.

Usaha yang besar dalam transformasi data (*data transformation*) adalah peningkatan kualitas data. Secara sederhana, ini mencakup antara lain pengisian nilai-nilai yang hilang dari atribut-atribut dalam data yang telah diekstrak. Kualitas data adalah hal yang paling penting dalam *data warehouse*, karena akibat yang ditimbulkan dari keputusan-keputusan strategis yang diambil berdasarkan pada informasi yang salah dapat sangat merugikan.

Fungsi-fungsi transformasi data dapat dibedakan menjadi beberapa tugas dasar, diantaranya:

a. Selection

Terletak di awal proses transformasi data. Yaitu memilih seluruh atau bagian-bagian dari beberapa *record* dari *source systems*.

b. Splitting/joining

Tugas ini meliputi manipulasi data yang diperlukan untuk bagian-bagian *record* hasil operasi *selection*. Kadang-kadang data akan dipisahkan (*split*) bahkan lebih jauh lagi selama transformasi data. Operasi *join* yang dilakukan terhadap bagian-bagian dari *record* hasil operasi *selection* lebih banyak terjadi di dalam *data warehouse*.

c. *Conversion*

Tahap ini meliputi konversi dari sebuah *field*, untuk dua alasan utama. Pertama untuk standarisasi data-data hasil ekstraksi dari *source systems* yang berbeda-beda, dan yang kedua untuk membuat *field-field* berarti dan dapat dimengerti oleh user.

d. *Summarization*

Terkadang tidak memungkinkan untuk menyimpan data pada detail level terendah dalam *data warehouse*. Mungkin user tidak memerlukan data pada granularitas terendah untuk analisis atau *query*, karena itu diperlukan ringkasan (*summary*) untuk disimpan ke dalam *data warehouse*.



e. *Enrichment*

Tugas ini adalah mengatur ulang dan menyederhanakan *field* individual untuk membuatnya lebih berguna untuk lingkungan *data warehouse*.

Tipe-tipe yang transformasi data yang paling umum antara lain:

a. *Format Revisions*

Revisi-revisi ini meliputi perubahan tipe data dan panjang data dari *field-field* individual. Pada *source systems*, data mungkin mempunyai tipe yang berbeda untuk *field* data yang sama. Lagipula panjang datanya juga mungkin berbeda-beda. Langkah yang tepat adalah melakukan standarisasi dan mengubah tipe data ke dalam bentuk teks untuk dapat lebih dipahami oleh user.

b. *Decoding of Fields*

Ini juga merupakan tipe transformasi data yang umum. Pada *source system* yang berbeda mungkin terdapat *field-field* yang mengandung kode-kode yang berbeda pula untuk merepresentasikan nilai dari *field tersebut*. Contohnya sebuah *source system* yang menggunakan angka 1 dan 2 untuk menandakan laki-laki dan perempuan. Kode ini tidak akan mempunyai arti untuk *end-user*, karena itu perlu diubah ke bentuk yang dapat dimengerti oleh *end-user*.

c. *Calculated and Derived Values*

Tipe ini berguna untuk menghitung dan mendapatkan nilai-nilai yang diperlukan dalam *data warehouse*. Seperti misalnya pendapatan harian rata-rata dan rasio operasi.

d. *Splitting of Single Fields*

Contoh untuk transformasi data tipe ini misalnya memisahkan komponen nama dan alamat ke dalam *field-field* yang berbeda di dalam *data warehouse*. Ini dilakukan untuk 2 alasan. Pertama untuk meningkatkan performa operasi dengan *indexing* dari komponen-komponen individual. Kedua, user mungkin ingin melakukan analisis dengan menggunakan komponen-komponen individual seperti kota, kabupaten, dan kode pos.

e. *Merging of Informations*

Tipe ini tidak merupakan kebalikan dari *Splitting of Single Fields*. Data transformation tipe ini tidak secara harafiah berarti menggabungkan beberapa *field* untuk membuat sebuah *field* data. Sebagai contoh, informasi tentang sebuah produk bisa berasal dari sumber-sumber data yang berbeda. Kode produk dan deskripsinya mungkin berasal dari *data source* yang sama, sedangkan tipe-tipe paket yang relevan dapat ditemukan dalam *data source* yang berbeda. Dalam kasus ini, menggabungkan informasi menunjukkan kombinasi dari kode produk, deskripsi, dan tipe-tipe paket ke dalam sebuah entitas.

f. *Character Set Conversions*

Data transformasi tipe ini berhubungan dengan konversi dari set karakter menjadi set karakter standar yang telah disetujui untuk data tekstual dalam *data warehouse*. Misalnya, mengkonversi data dari *source systems* dengan karakter-karakter EBCDIC ke dalam format ASCII.

g. *Conversions of Units Measurements*

Data transformasi tipe ini meliputi pengubahan ukuran sesuai dengan standar ukuran yang telah ditetapkan. Sebagai contoh jika sebuah perusahaan mempunyai cabang-cabang di beberapa negara yang menggunakan ukuran panjang berbeda (meter, kaki) maka dalam *data warehouse* harus ditentukan satu standar ukuran panjang untuk semua data.

h. *Date/Time Conversions*

Tipe ini berhubungan dengan penyajian tanggal dan waktu dalam format-format standar. Sebagai contoh adalah format tanggal Amerika dan Inggris yang akan distandarkan. Tanggal 11 Oktober tahun 2000 dalam format amerika akan tertulis 10/11/2000 dan dalam format Inggris 11/10/2000. Tanggal ini harus distandarkan dan ditulis menjadi 11 OCT 2000.

i. *Summarization*

Tipe transformasi ini adalah untuk membuat ringkasan (*summary*) untuk di-load ke dalam *data warehouse*, dan bukannya me-load level yang paling kecil dari data. Sebagai contoh, sebuah perusahaan *credit card* yang ingin menganalisa pola penjualan, mungkin tidak perlu untuk menyimpan setiap transaksi tunggal untuk tiap-tiap *credit card* ke dalam *data warehouse*. Sebaliknya hanya perlu dilakukan membuat *summary* dari transaksi-transaksi harian dari tiap-tiap *credit card* dan menyimpan *summary* dari data tersebut, daripada menyimpan data yang paling lengkap dengan transaksi-transaksi individual.

j. Key Restructing

Primary keys dari *data sources* yang diekstrak akan menjadi dasar dari kunci-kunci tabel dimensi dan fakta dalam *data warehouse*.

k. Deduplication

Contoh untuk tipe ini adalah data pada *file* pelanggan. Banyak *file-file* pelanggan yang mempunyai beberapa *record* untuk pelanggan yang sama. Seringkali duplikasi disebabkan karena penambahan *record-record* akibat kesalahan. Dalam *data warehouse* diperlukan sebuah *record* untuk seorang pelanggan, yang mana kemudian semua duplikat di *source systems* harus dihubungkan ke *record* tunggal ini. Proses ini disebut deduplikasi pada *file* pelanggan.

3. Data Loading

Setelah data ditransformasikan, langkah berikutnya adalah *data loading*. Sebagian besar *data loading* mencakup pengambilan data yang telah siap (bersih), mengaplikasikannya ke *data warehouse*, dan menyimpannya ke dalam *database* yang ada disana.

Keseluruhan proses memindahkan data ke dalam *repository data warehouse* dapat dilakukan dalam beberapa cara:

a. *Initial Load*

Mengumpulkan tabel-tabel *data warehouse* untuk yang pertama kalinya.

b. *Incremental Load*

Melakukan perubahan-perubahan secara terus-menerus seperlunya dalam kurun waktu tertentu (secara periodik).

c. *Full Refresh*

Menghapus seluruhnya isi tabel dan melakukan *reload* dengan data-data baru (*Initial Load* adalah merefresh seluruh tabel-tabel).

3.4.4.3 Komponen Data Storage

Penyimpanan data untuk intelegensi bisnis diletakkan pada tempat penyimpanan (*repository*) yang berbeda. Tempat penyimpanan tersebut berupa data warehouse, data mart, maupun multidimensional data. Diperlukan tempat penyimpanan yang terpisah dari data sistem operasional.

Pada umumnya suatu sistem operasional dalam perusahaan mempunyai tempat penyimpanan untuk sistem operasional yang hanya mengandung *current data* saja. Penyimpanan data untuk suatu *data warehouse* digunakan untuk menyimpan data historikal yang bervolume besar yang untuk melakukan suatu analisis. Data dalam *data warehouse* hanyalah bersifat *read only*, sedangkan penyimpanan data untuk sistem operasional biasanya digunakan dalam suatu proses transaksional sehingga data-data yang terdapat didalamnya dapat terus diupdate.

3.4.4.4 Komponen Information Delivery

Pengguna yang memerlukan informasi dari *data warehouse* antara lain:

1. Pengguna baru: menggunakan *data warehouse* tanpa pelatihan dan karena itu memerlukan *template-template report* dan *query-query* yang telah di-set sebelumnya.
2. Pengguna tidak tetap: memerlukan informasi hanya sesekali, tidak secara taratur. Pengguna tipe ini juga memerlukan paket informasi yang dipersiapkan.
3. Pengguna analis bisnis: memerlukan fasilitas untuk melakukan analisis kompleks dengan informasi dalam *data warehouse*.
4. Pengguna yang berkemampuan: ingin dapat melihat-lihat/mengetahui seluruh *data warehouse*, mengambil data-data yang menarik, membentuk *query-query* sendiri, melakukan *drill* pada lapisan-lapisan data, dan membuat *report-report* tertentu dan *query-query ad hoc*.

Dalam usaha menyediakan informasi untuk komunitas pengguna-pengguna *data warehouse*, komponen *information delivery* meliputi beberapa metode untuk melakukan penyampaian informasi.

Dalam *data warehouse* mungkin diperlukan untuk memasukkan lebih dari satu mekanisme *information delivery*. Yang paling umum yaitu penyediaan *query-query* dan *report-report* secara *online*. Para *user* akan menyampaikan permintaan mereka secara *online* dan menerima hasil-hasilnya secara *online* pula. Dapat juga ditetapkan penyampaian *report-report* terjadwal melalui e-mail, atau dapat digunakan intranet milik organisasi yang memadai untuk penyampaian informasi. Bahkan sekarang ini penyampaian informasi melalui internet pun sudah mulai meluas.

3.4.4.5 Komponen Metadata

Metadata dalam suatu *data warehouse* mirip dengan kamus data atau katalog data dalam sebuah *database management systems*. Dalam kamus data, informasi yang disimpan yaitu mengenai struktur data yang bersifat logika, informasi mengenai *file* dan alamat, informasi mengenai indeks dan lain sebagainya. Kamus data berisi data mengenai data dalam *database*.

Sama halnya dengan kamus data, komponen metadata adalah data mengenai data dalam *data warehouse*. Metadata dalam *data warehouse* dibagi menjadi 3 kategori utama:

1. Operational Metadata

Seperti yang telah diketahui, data untuk *data warehouse* datang dari beberapa sistem operasional milik perusahaan. *Source systems* ini memiliki struktur data yang berbeda-beda. Dalam memilih data dari *source systems* untuk *data warehouse*, dilakukan tahap pembersihan data. Ketika informasi disampaikan kepada *end-user*, informasi tersebut

harus dapat dihubungkan kembali ke data set-data set sumber. *Operational metadata* mengandung semua informasi tentang sumber-sumber data operasional.

2. **Extraction and Transformation Metadata**

Extraction and transformation metadata mengandung data tentang ekstraksi data dari *source systems* yaitu frekuensi ekstraksi, metode-metode ekstraksi, dan aturan-aturan bisnis untuk ekstraksi data. Kategori metadata ini juga mengandung informasi tentang semua transformasi data yang berlangsung di staging area.

3. **End-User Metadata**

End-user metadata adalah peta dari *data warehouse* yang memungkinkan *end-user* untuk menemukan informasi dalam *data warehouse*. *End-user metadata* memungkinkan *end-user* untuk menggunakan terminologi bisnis mereka sendiri dan mencari informasi dalam cara-cara yang seperti biasanya mereka lakukan dalam bisnis.

3.4.4.6 **Komponen Management and Control**

Komponen ini dalam arsitektur *data warehouse* menempati peringkat atas. Komponen manajemen dan kontrol mengatur *service-service* dan aktivitas-aktivitas dalam *data warehouse*. Komponen ini juga mengendalikan transformasi data dan transfer data ke dalam penyimpanan *data warehouse*, serta mengatur penyampaian informasi ke user. Komponen ini bekerjasama dengan DBMS dan memungkinkan data untuk dapat disimpan dengan baik di *repository-repository*. Tugasnya adalah memonitor perpindahan data ke dalam *staging area*, dan dari *staging area* ke dalam penyimpanan *data warehouse* itu sendiri. Komponen kontrol dan metadata berinteraksi dengan komponen metadata untuk melakukan fungsi-fungsi kontrol dan manajemen. Karena komponen metadata mengandung informasi tentang *data warehouse* itu sendiri, maka metadata adalah sumber informasi untuk modul manajemen.

3.5 **Data Mart**

Selain *data warehouse*, dikenal juga istilah *data mart*. *Data mart* merupakan bentuk kecil atau subset dari *data warehouse* yang menyediakan informasi-informasi spesifik untuk kebutuhan-kebutuhan departemental bagi pengguna. Jadi *data mart* dapat berupa *data warehouse* yang berorientasi subjek untuk keperluan-keperluan fungsional atau departemental, atau dapat juga berupa *enterprisewide data warehouse* dalam bentuk kecil yang mengkombinasikan data dari beberapa area subjek dan bertindak sebagai sebuah titik sumber dari *enterprise data warehouse*.

3.5.1 **Perbedaan Data Warehouse dan Data Mart**

Perbedaan-perbedaan antara *data mart* dan *data warehouse* dapat dilihat pada tabel di bawah ini:

3.2 Perbedaan Data Warehouse dan Data Mart (Green, 2003)

Properti	<i>Data Warehouse</i>	<i>Data Mart</i>
Lingkup	Perusahaan (<i>Enterprise</i>)	Departemen
Subjek	Lebih dari satu	Subjek tunggal, <i>Line of Business(LOB)</i>
Sumber Data	Banyak	Sedikit
Ukuran	GB sampai TB	Ratusan MB sampai GB
Waktu Implementasi	Bulanan sampai tahunan	Bulanan

1. **Lingkup**

Sebuah *data warehouse* berhubungan dengan lebih dari satu area subjek dan biasanya diimplementasikan dan diatur oleh sebuah unit organisasional pusat seperti departemen IT perusahaan. Seringkali disebut dengan *data warehouse* pusat atau perusahaan.

Sedangkan *data mart* biasanya hanya dibuat untuk departemen atau bagian dari perusahaan yang tertentu saja dan tidak mewakili seluruh informasi perusahaan seperti *data warehouse*.

2. Subjek

Sebuah *data mart* merupakan bentuk departemental dari *data warehouse* yang dirancang untuk sebuah garis bisnis tunggal (*single line of business/LOB*).

3. Sumber data

Sebuah *data warehouse* umumnya mengambil data dari banyak sistem sumber, sedangkan *data mart* mengambil data dari sumber-sumber yang jumlahnya lebih sedikit.

4. Ukuran

Data mart tidak dibedakan dari *data warehouse* berdasarkan ukuran, tetapi dalam penggunaan dan manajemen. Satu definisi dari *warehouse* yang sangat besar adalah: "Suatu *warehouse* yaitu lebih besar daripada *backup time window*."

5. Waktu implementasi

Data mart biasanya lebih sederhana daripada *data warehouse* dan karena itu lebih mudah untuk dibuat dan dipelihara. Sebuah *data mart* juga dapat dibuat sebagai langkah "pembuktian konsep" terhadap pembangunan sebuah *enterprisewide data warehouse*.

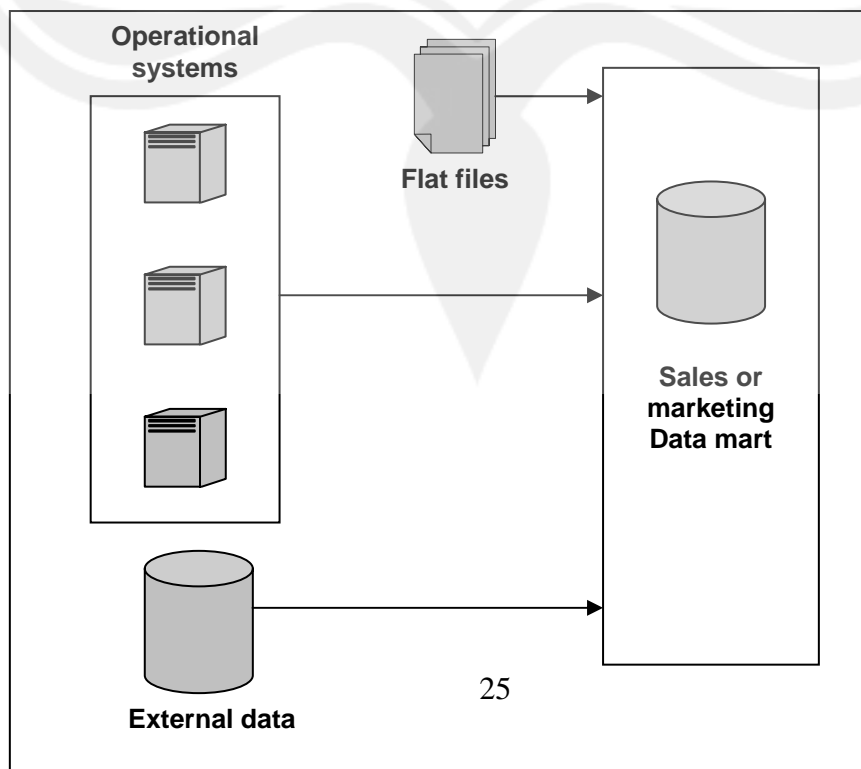
3.5.2 Tipe-Tipe Data Mart

Pengkategorian ini didasarkan pada sumber data dari *data mart* tersebut. *Data mart* dapat dikategorikan ke dalam dua tipe:

1. Independent Data Mart

Independent data mart merupakan sebuah sistem *stand-alone* yang dibangun per-bagian, yang mengambil data langsung dari sumber-sumber data operasional atau eksternal. *Independent data mart* mempunyai karakteristik sebagai berikut:

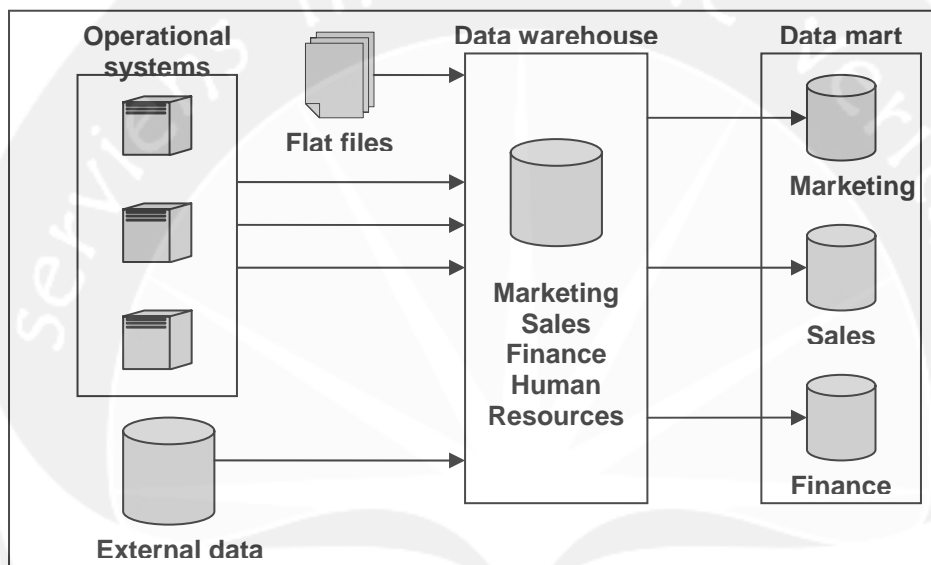
- Sumber-sumbernya adalah sistem-sistem operasional dan eksternal.
- Proses ETL-nya sulit. Karena *independent data mart* mengambil data dari sumber-sumber data yang belum bersih atau belum konsisten, sehingga usaha yang dilakukan dititikberatkan kepada *error processing* dan integritas data.
- Data mart* dibangun untuk memenuhi kebutuhan-kebutuhan analitik. Penciptaan *independent data mart* seringkali diakibatkan karena kebutuhan solusi yang cepat untuk permintaan-permintaan analisis.



2. *Dependent Data Mart*

Dependent *data mart* mempunyai karakteristik-karakteristik sebagai berikut:

- Sumber datanya adalah *data warehouse*. Dependent *data mart* mengandalkan *data warehouse* untuk isinya.
- Proses ETL (*Extraction, Transformation, Load*)–nya mudah. Dependent *data mart* mengambil data dari *data warehouse* pusat yang telah terlebih dahulu dibangun. Karena itu upaya utama dari membangun sebuah *data mart*, yaitu membersihkan dan mengekstraksi, dapat dilewatkan. *Data mart* tipe ini hanya membutuhkan data untuk dipindahkan dari satu *database* ke *database* lainnya.
- Data mart* ini adalah bagian dari perencanaan perusahaan. Dependent *data mart* biasanya dibangun untuk mencapai ketersediaan data dan performa yang lebih baik, kontrol yang lebih baik, dan biaya telekomunikasi yang lebih rendah sebagai hasil dari akses lokal ke data yang relevan terhadap departemen tertentu.



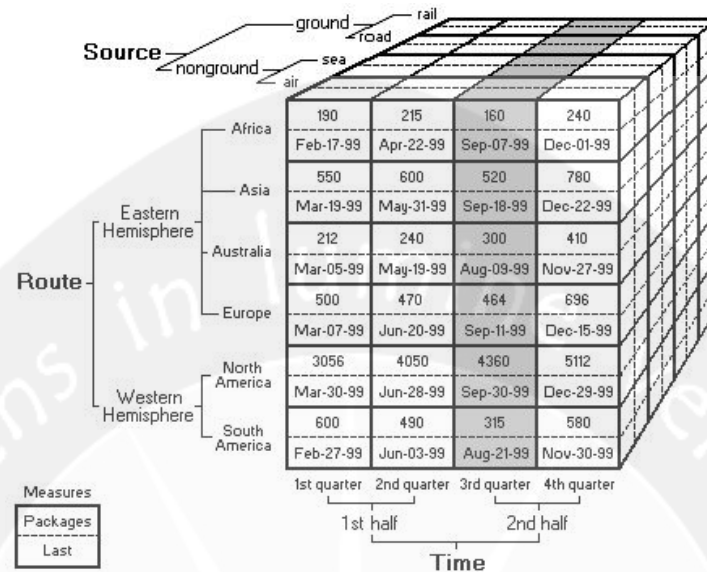
Gambar 3.7. *Dependent Data Mart* (Green, 2003)

3.6 Model Data Multidimensional

Model data multidimensional adalah himpunan pengukuran numerik yang tergantung pada himpunan dimensi, misalnya untuk mengetahui Penjualan, dimensinya adalah Produk, Lokasi, dan Waktu. Model data multidimensional merupakan rancangan logikal yang bertujuan untuk menampilkan data dalam bentuk standar dan intuitif yang memperbolehkan akses dengan performa yang tinggi. Model data multidimensional dirancang untuk memfasilitasi analisis dan bukan transaksi. Model ini umum digunakan dalam *data warehouse*. Memiliki konsep intuitif dari banyak dimensi atau perspektif pengukuran bisnis atau fakta-fakta. Contohnya: untuk melihat penjualan dari perspektif pelanggan, produk dan waktu.

Model multidimensional menggunakan konsep model hubungan antar *entity* (ER) dengan beberapa batasan yang penting. Setiap model multidimensional terdiri dari sebuah tabel dengan sebuah komposit *primary key* dan merupakan relasi utama yang berhubungan dengan dimensi yang diukur, disebut dengan tabel fakta, dan satu set tabel yang lebih kecil disebut tabel dimensi. Tiap dimensi dapat diberi tambahan atribut dan berasosiasi dengan suatu tabel dimensi. Tabel fakta mempunyai ukuran yang lebih besar dibandingkan dengan tabel dimensi. Unit pemrosesan data yang terdiri dari tabel fakta dan dimensi dalam suatu *data warehouse* disebut kubus data (*cube*). Contohnya suatu kubus data seperti Sales memungkinkan data untuk dimodelkan dan dilihat dari banyak dimensi:

- Tabel dimensi, seperti item (item_name, brand, type), atau time(day, week, month, quarter, year)
- Tabel fakta memuat ukuran (seperti dollars_sold) dan kunci untuk setiap dimensi tabel terkait

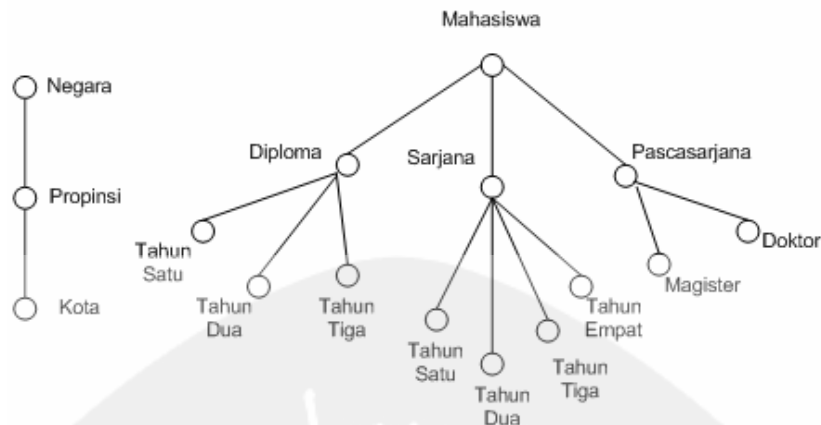


Gambar 3.8. Kubus Data (Cube)
(Sumber : biolap.sourceforge.net)

3.6.1 Konstruksi Model Data Multidimensional

Pada dasarnya, fakta, dimensi dan hirarki dimensi adalah tiga konstruksi model multidimensional yang umum ditemukan dalam perancangan *data warehouse*:

- Fakta adalah peristiwa atau proses yang terjadi secara dinamik dalam dunia organisasi untuk menghasilkan data sepanjang waktu. Fakta dapat dipandang sebagai sebuah entitas transaksi yang mengandung pengukuran atau kuantitas dan dapat diringkaskan melalui berbagai dimensi. Pengukuran atau nilai kuantifikasi merupakan fokus perhatian bagi proses pengambilan keputusan.
- Dimensi adalah objek-objek yang dihubungkan melalui asosiasi yang berfungsi sebagai konteks kualifikasi dan terstruktur menurut satu atau lebih jalur agregat yang berkongsi level dimensi akhir. Dimensi berasal dari atribut-atribut diskrit yang menentukan butiran-butiran fakta minimum dan dikategorikan secara sintaksis guna menetapkan cara-cara untuk melihat informasi, sesuai dengan perspektif alamiah bisnis dimana analisa faktanya dapat dilakukan.
- Hirarki dimensi terbentuk dari atribut-atribut diskrit dimensi yang dihubungkan oleh asosiasi dan menentukan bagaimana fakta dapat disusun dan dipilih secara signifikan untuk proses pengambilan keputusan. Hirarki dimensi dapat diklasifikasikan ke dalam dua jenis dasar (Akoka et al. 2001), yaitu :
 - Hirarki sederhana hanyaterdiri dari satu jalur agregat linier di dalam sebuah dimensi, misalnya kota → propinsi → negara.
 - Hirarki majemuk, terdiri dari sekurang-kurangnya dua jalur agregat berbeda dalam sebuah dimensi. Contoh: dimensi mahasiswa di sebuah domain universitas terdiri dari mahasiswa diploma, sarjana dan pascasarjana. Mahasiswa sarjana terdiri dari empat kelas yang menunjukkan tahun si mahasiswa di universitas, sedangkan mahasiswa pascasarjana boleh mahasiswa program magister atau mahasiswa program doktor.



Gambar 3.9. Hirarki Dimensi (Sitompul,2008)

3.6.2 Skema Data Multidimesional

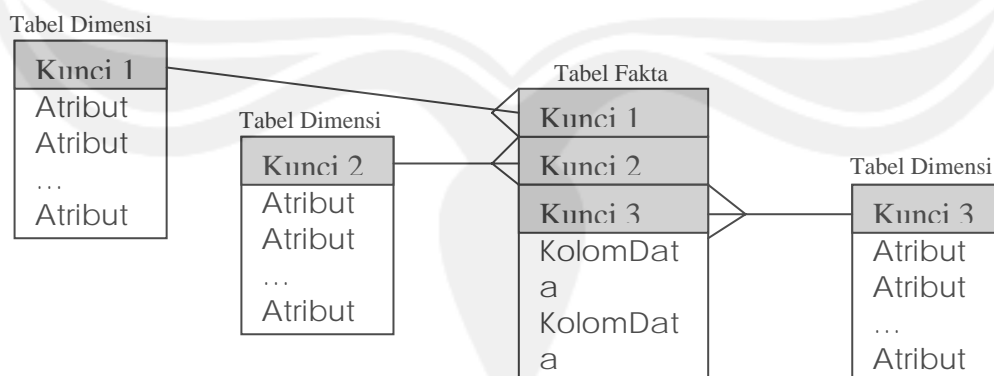
Sebuah sistem OLTP memerlukan normalisasi untuk mengurangi redundansi, validasi untuk input data, mendukung volume yang besar dari transaksi yang bergerak sangat cepat. Model OLTP sering terlihat seperti jaring laba-laba yang terdiri atas ratusan bahkan ribuan tabel sehingga sulit untuk dimengerti. Sebaliknya, model dimensional yang sering digunakan pada *data warehouse* adalah skema bintang atau *snowflake* yang mudah dimengerti dan sesuai dengan kebutuhan bisnis, mendukung *query* sederhana dan menyediakan performa *query* yang superior dengan meminimalisasi tabel-tabel join.

1. Skema Bintang

Skema bintang merupakan struktur logikal yang memiliki tabel fakta yang terdiri atas data faktual di tengahnya, dan dikelilingi oleh tabel-tabel dimensi yang berisi referensi data. Setiap tabel dimensi memiliki sebuah *simple primary key* yang merespon tepat pada satu komponen *primary key* pada tabel fakta. Dengan kata lain *primary key* pada table fakta terdiri dari dua atau lebih *foreign key*. Jenis-jenis skema bintang:

a. Skema bintang sederhana

Dalam skema ini, setiap tabel harus memiliki *primary key* yang terdiri dari satu kolom atau lebih. *Primary key* dari tabel fakta terdiri dari satu atau lebih *foreign key*. *Foreign key* merupakan *primary key* pada tabel lain.

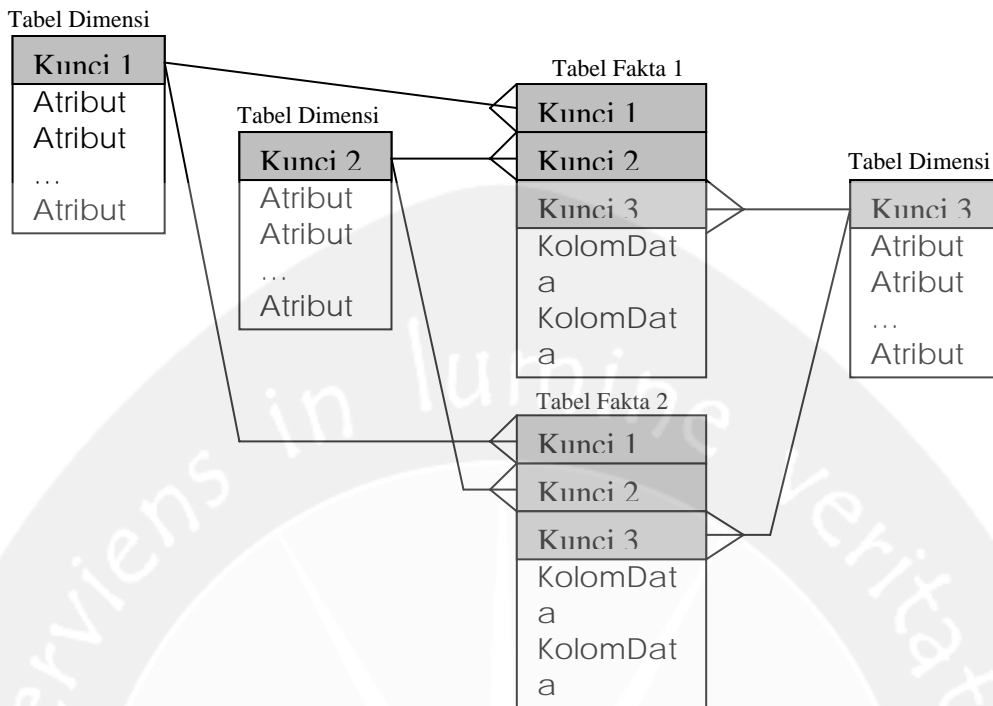


Gambar 3.10. Skema Bintang Sederhana

b. Skema bintang dengan banyak tabel fakta

Skema bintang juga bisa terdiri dari satu atau lebih tabel fakta. Dikarenakan karena tabel fakta tersebut ada banyak, misalnya selain penjualan terdapat tabel fakta

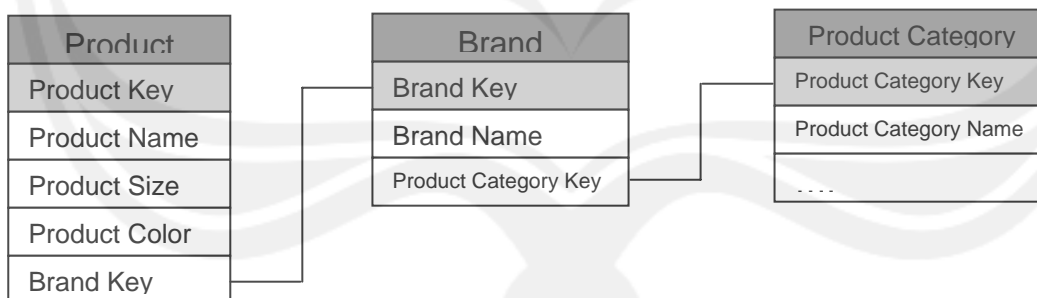
forecasting dan *result*. Walaupun terdapat lebih dari satu tabel fakta, mereka tetap menggunakan tabel dimensi bersama-sama.



Gambar 3.11. Skema Bintang dengan Banyak Tabel Fakta

2. Snowflake Schema

Merupakan varian dari skema bintang dimana tabel-tabel dimensi tidak terdapat data yang didenormalisasi. Dengan kata lain satu atau lebih tabel dimensi tidak bergabung secara langsung kepada tabel fakta tapi pada tabel dimensi lainnya. Sebagai contoh, sebuah dimensi yang mendeskripsikan produk dapat dipisahkan menjadi tiga table (*snowflaked*) seperti pada gambar 3.12.



Gambar 3.12. Skema Snowflake

Keduanya skema ini merupakan model-model dimensional, perbedaannya terletak pada implementasi fisik. Skema *snowflake* memberi kemudahan pada perawatan dimensi, dikarenakan strukturnya yang lebih normalisasi. Sedangkan skema bintang lebih efisien serta sederhana dalam membuat *query* dan mudah diakses secara langsung oleh pengguna.

BAB IV

ANALISIS DAN PERANCANGAN

DATA WAREHOUSE DAN BUSINESS INTELLIGENCE

Pada bab ini akan dilakukan analisis dan perancangan tabel dimensi dan tabel fakta untuk *data mart* untuk subjek kegiatan akademik di Universitas Atma Jaya Yogyakarta.

4.1 Analisis Kebutuhan Datawarehouse

Setiap tahunnya masing-masing program studi melakukan laporan akademik ke Koordinasi Perguruan Tinggi Swasta (Kopertis) dalam bentuk laporan EPSBED (Evaluasi Program Studi Berbasis Evaluasi Diri) dan setiap lima tahun sekali pihak universitas memberikan data akademik ke Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) untuk keperluan akreditasi. Untuk memudahkan pihak universitas dalam menyusun laporan, memonitor kegiatan akademik dan membantu dalam proses pengambilan keputusan maka aplikasi intelegensi bisnis diperlukan. Aplikasi ini akan bermanfaat dalam pengurangan biaya dalam pembuatan laporan-laporan, aktivitas yang membutuhkan informasi secara cepat dan akurat, dan secara signifikan dapat meningkatkan kualitas keputusan yang dibuat oleh pihak universitas.

Dalam penelitian ini, kasus yang dianalisis adalah kegiatan akademik di Universitas Atma Jaya Yogyakarta. Karena sistem pendukung keputusan yang akan dibangun dikhususkan pada subjek tertentu, maka sistem ini bukan merupakan *enterprisewide data warehouse* dari universitas melainkan sebuah *data mart departemental* dengan fungsionalitas untuk menyediakan informasi-informasi yang dibutuhkan dalam lingkup kegiatan akademik.

Analisis kebutuhan untuk keperluan *data mart* yang akan dibuat merupakan *independent data mart* karena mengambil data langsung dari sumber-sumber data operasional. Kebutuhan-kebutuhan data dan informasi yang diperlukan untuk pelaporan didapatkan dari berbagai sumber seperti borang akreditasi program studi, borang akreditasi universitas, dan dokumen-dokumen lain. Informasi-informasi tersebut kemudian dibatasi untuk subjek akademik, yaitu :

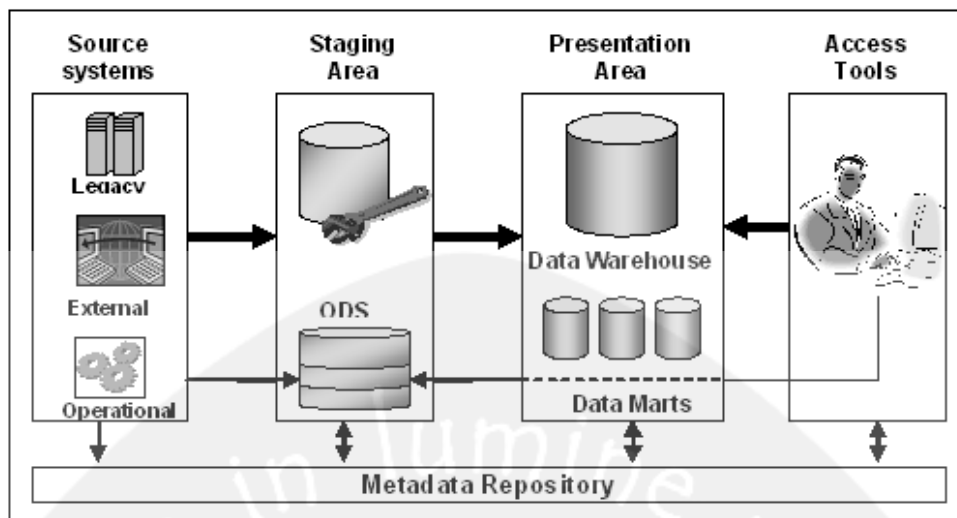
1. Bagaimanakah profil status mahasiswa pada semua prodi?
2. **Bagaimanakah profil status mahasiswa berdasarkan tahun angkatan dan jenis kelamin pada prodi di lingkungan Universitas Atma Jaya Yogyakarta ?**
3. Berapakah jumlah mahasiswa *drop out* berdasarkan tahun angkatan pada semua prodi?
4. Berapakah jumlah mahasiswa *drop out* berdasarkan tahun angkatan dan jenis kelamin pada prodi di lingkungan Universitas Atma Jaya Yogyakarta ?
5. Bagaimanakah perbandingan jumlah mahasiswa yang mengambil konsentrasi tertentu berdasarkan tahun angkatan pada prodi Teknik Informatika Fakultas Teknologi Industri?
6. Bagaimanakah perbandingan jumlah mahasiswa yang mengambil konsentrasi tertentu berdasarkan jenis kelamin pada prodi Teknik Informatika Fakultas Teknologi Industri?
7. Bagaimanakah perbandingan IPK mahasiswa pada semua prodi pada tahun 2008, 2009, dan 2010?
8. Bagaimanakah perbandingan rata-rata IPK mahasiswa berdasarkan tahun angkatan pada semua prodi pada semester gasal tahun ajaran 2009/2010?
9. Bagaimanakah perbandingan IPK mahasiswa berdasarkan tahun angkatan dan jenis kelamin pada prodi Teknik Informatika pada semester gasal tahun ajaran 2009/2010?

10. Bagaimanakah perbandingan IPS mahasiswa pada semua prodi pada tahun 2008, 2009, dan 2010?
11. Bagaimanakah perbandingan rata-rata IPS mahasiswa berdasarkan tahun angkatan pada semua prodi pada semester gasal tahun ajaran 2009/2010?
12. Bagaimanakah perbandingan IPS mahasiswa berdasarkan tahun angkatan dan jenis kelamin pada prodi Teknik Informatika pada semester gasal tahun ajaran 2009/2010?
13. Bagaimanakah perbandingan pengambilan sks mahasiswa pada semua prodi pada tahun 2008, 2009, dan 2010?
14. Bagaimanakah perbandingan rata-rata pengambilan sks mahasiswa berdasarkan tahun angkatan pada semua prodi pada semester gasal tahun ajaran 2009/2010?
15. Bagaimanakah perbandingan pengambilan sks mahasiswa berdasarkan tahun angkatan dan jenis kelamin pada prodi Teknik Informatika pada semester gasal tahun ajaran 2009/2010?
16. Bagaimanakah perbandingan jumlah kelulusan berdasarkan IPK dan tahun angkatan pada prodi Teknik Informatika Fakultas Teknologi Industri pada tahun 2008, 2009, dan 2010?
17. Bagaimanakah perbandingan jumlah kelulusan berdasarkan IPK dan jenis kelamin pada semua prodi pada tahun 2009?
18. Bagaimanakah perbandingan jumlah kelulusan berdasarkan masa studi dan tahun angkatan pada prodi Teknik Informatika Fakultas Teknologi Industri pada tahun 2008, 2009, dan 2010?
19. Bagaimanakah perbandingan jumlah kelulusan berdasarkan masa studi dan jenis kelamin pada semua prodi pada tahun 2009?
20. Bagaimanakah sebaran nilai matakuliah Rekayasa Perangkat Lunak pada prodi Teknik Informatika pada tahun 2008, 2009, dan 2010?
21. Bagaimanakah tingkat perolehan nilai untuk matakuliah 3 sks pada prodi Teknik Informatika Fakultas Teknologi Industri pada semester gasal tahun ajaran 2009/2010?
22. Bagaimanakah tingkat kelulusan matakuliah berdasarkan sifat matakuliah pada prodi Teknik Informatika Fakultas Teknologi Industri pada semester gasal tahun ajaran 2009/2010?
23. Bagaimanakah tingkat kelulusan matakuliah Kerja Praktek pada prodi Teknik Informatika pada tahun 2008, 2009, dan 2010?
24. Bagaimanakah perbandingan jumlah beban dosen semua dosen pada prodi Teknik Informatika Fakultas Teknologi Industri tahun 2008, 2009, dan 2010?
25. Bagaimanakah perbandingan jumlah beban dosen berdasarkan jenis dosen pada semua prodi pada tahun 2008, 2009, dan 2010?
26. Bagaimanakah perbandingan jumlah beban dosen berdasarkan jenis kelamin pada prodi Teknik Informatika Fakultas Teknologi Industri tahun 2008, 2009, dan 2010?

4.2 Proses Pembangunan Aplikasi Business Intelligence

Aplikasi Business Intelligence dimulai dari proses ekstraksi data dari sumber data (*source systems*). Data tersebut adalah sumber-sumber data yang dibutuhkan oleh *data mart*; *staging area* adalah tempat dimana data dibersihkan dan disiapkan dengan proses-proses ETL; *presentation area* adalah dimana data disimpan dan dioptimalkan untuk *query*, *reporting* dan analisis yang dapat berupa *data warehouse* atau *data mart*-*data mart*; sedangkan *access tools* digunakan oleh pengguna (*end-user*) untuk mengakses informasi yang ada di *presentation area*.

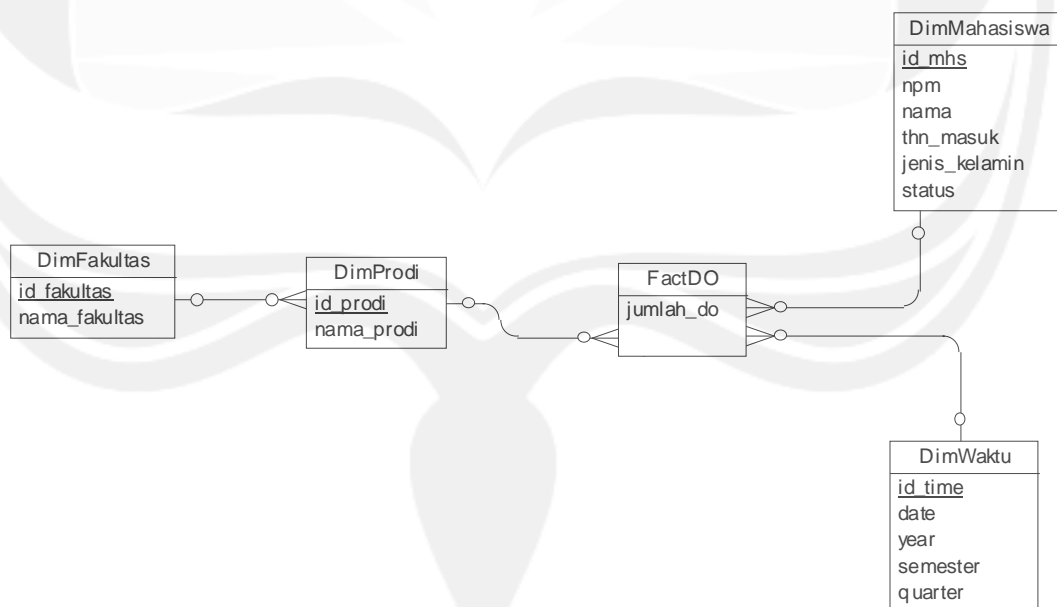
Dalam gambar 4.1. dapat dilihat komponen-komponen umum sebuah lingkungan *business intelligence* secara garis besar.



Gambar 4.1. Komponen-Komponen Umum Sistem Lingkungan *Data Warehouse/Mart*

Pada sub bab sebelumnya telah dilakukan analisis untuk mengetahui informasi-informasi apa saja yang harus disediakan oleh sistem. Dari analisis tersebut dibuat sebuah skema model data. Model data yang dibutuhkan untuk *datawarehouse* dirancang dengan model star schema, yang berikutnya akan digunakan untuk membangun data mart.

Skema star dengan beberapa ekstensi *snow flake* untuk profil status mahasiswa berdasar fakultas, program studi, jenis kelamin, dan status Drop Out tampak pada gambar berikut :



Gambar 4.2. Desain Star Schema untuk profile mahasiswa

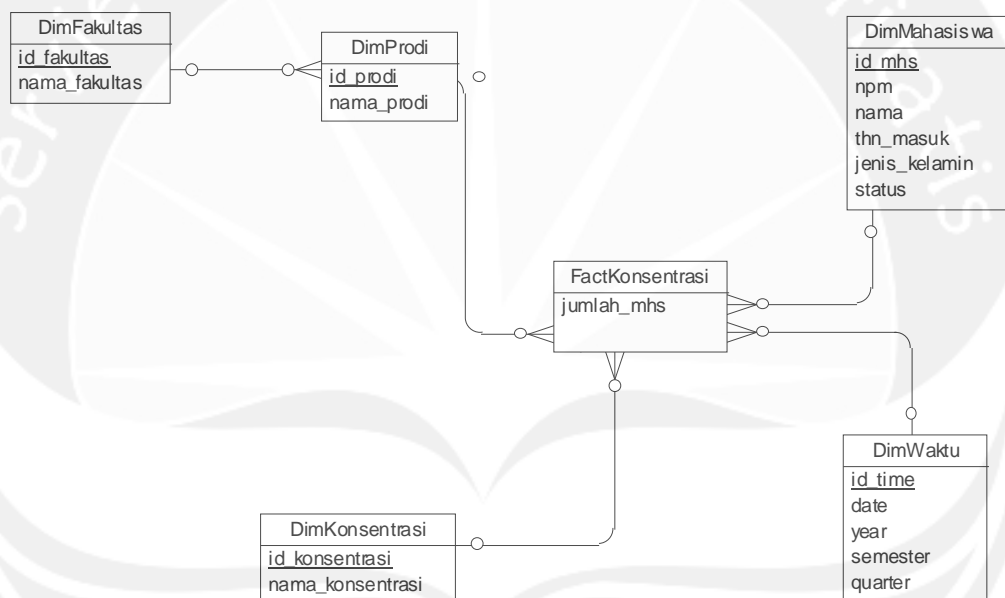
Information Package untuk subjek ini tampak pada tabel berikut :

Subjek Informasi : Profile Mahasiswa Drop Out

Dimensi

Waktu	Mahasiswa	Prodi	Fakultas	
Year	Tahun masuk	Nama Prodi	Nama Fakultas	
Semester	Jenis kelamin			
Quarter	Status			
Date	Nama			
	NPM			
Fact	JumlahDO			

Skema star dengan beberapa ekstensi *snow flake* untuk konsentrasi studi mahasiswa berdasar fakultas, program studi, dan jenis kelamin tampak pada gambar berikut :



Gambar 4.3. Desain Star Schema untuk konsentrasi studi mahasiswa

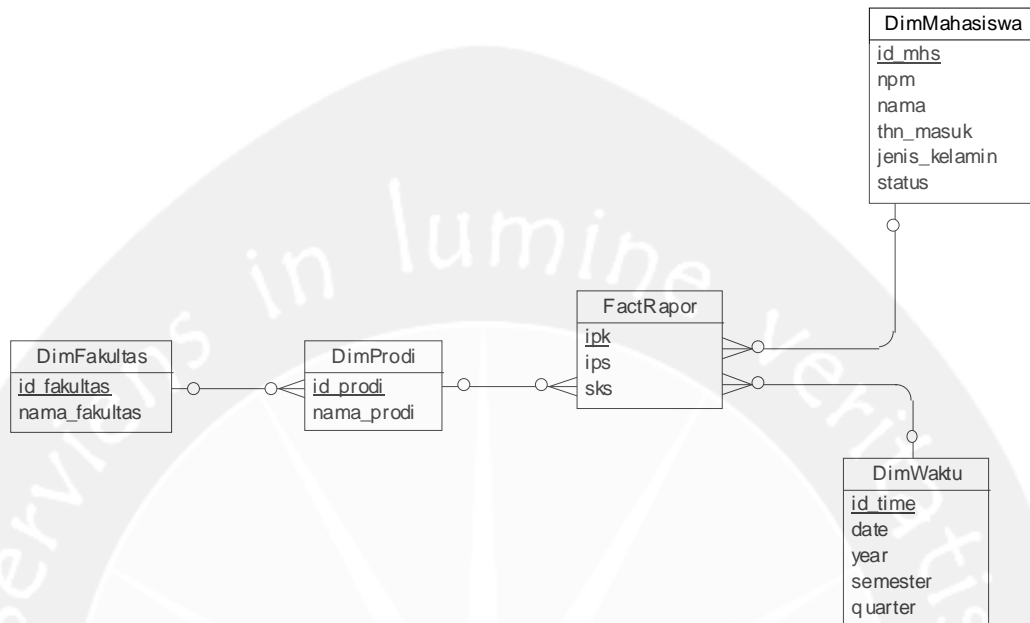
Information Package untuk subjek ini tampak pada tabel berikut :

Subjek Informasi : Profile Mahasiswa Konsentrasi

Dimensi

Waktu	Mahasiswa	Prodi	Fakultas	Konsentrasi
Year	Tahun masuk	Nama Prodi	Nama Fakultas	Jenis Konsentrasi
Semester	Jenis kelamin			
Quarter	Status			
Date	Nama			
	NPM			
Fact	Jumlah Mahasiswa			

Skema star dengan beberapa ekstensi *snow flake* untuk IPK, IPS dan SKS mahasiswa berdasar fakultas, program studi, dan jenis kelamin tampak pada gambar berikut :



Gambar 4.4. Desain Star Schema untuk IPK, IPS, dan SKS mahasiswa

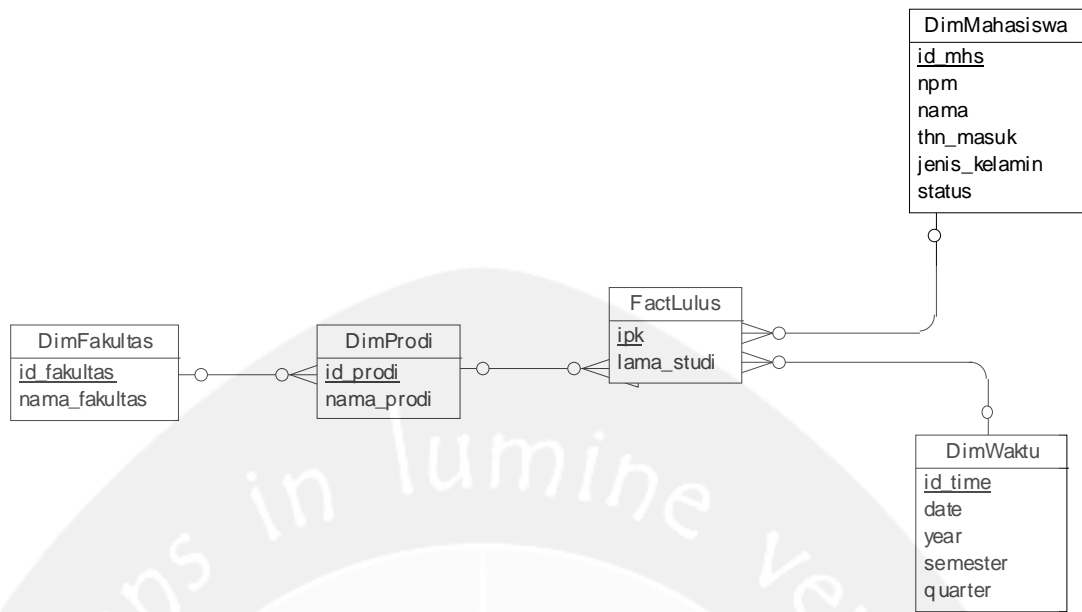
Information Package untuk subjek ini tampak pada tabel berikut :

Subjek Informasi : Rapor Mahasiswa

Dimensi

Waktu	Mahasiswa	Prodi	Fakultas	
Year	Tahun masuk	Nama Prodi	Nama Fakultas	
Semester	Jenis kelamin			
Quarter	Status			
Date	Nama			
	NPM			
Fact	IPK, IPS, SKS			

Skema star dengan beberapa ekstensi *snow flake* untuk IPK kelulusan mahasiswa berdasar fakultas, program studi, dan jenis kelamin tampak pada gambar berikut :



Gambar 4.5. Desain Star Schema untuk kelulusan mahasiswa

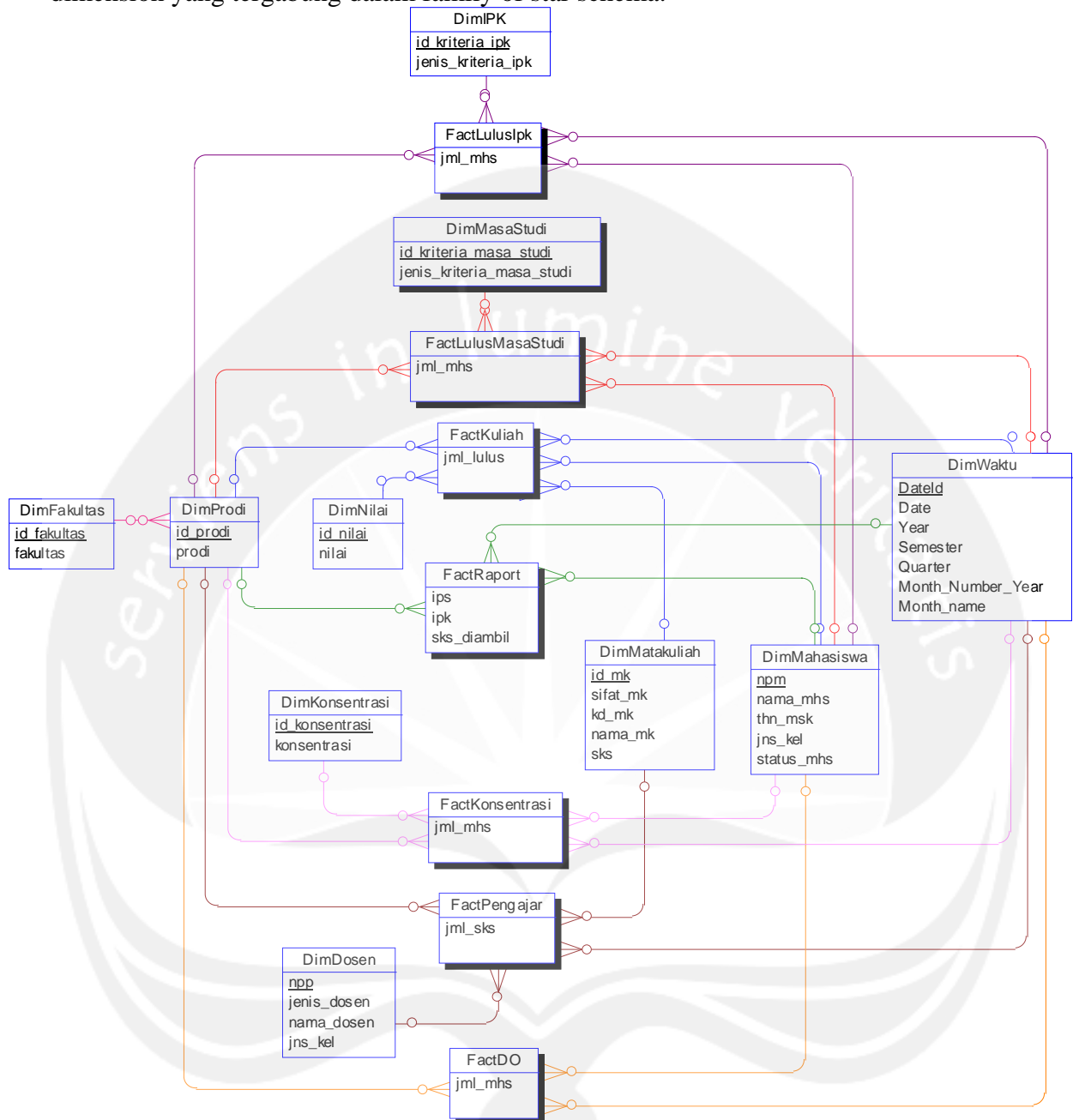
Information Package untuk subjek ini tampak pada tabel berikut :

Subjek Informasi : Kelulusan

Dimensi

Waktu	Mahasiswa	Prodi	Fakultas	
Year	Tahun masuk	Nama Prodi	Nama Fakultas	
Semester	Jenis kelamin			
Quarter	Status			
Date	Nama			
	NPM			
Fact	IPK, Lama Studi			

Secara keseluruhan desain Star Schema digabungkan dalam bentuk conformed dimension yang tergabung dalam family of star schema.



Gambar 4.6. Conformed Dimensions pada family of star schemas

BAB V

KESIMPULAN

Pada bab ini akan dituliskan kesimpulan dari penelitian ini.

5.1. Kesimpulan

1. Penelitian ini telah menghasilkan rancangan pengembangan Data Warehouse dan Aplikasi Intelegensi Bisnis untuk subjek akademik di Universitas Atma Jaya Yogyakarta. Analisis telah dilakukan dengan bersumber pada dokumen laporan EPSBED (Evaluasi Program Studi Berbasis Evaluasi Diri) dan dokumen borang akreditasi ke Badan Akreditasi Nasional Perguruan Tinggi (BAN-PT) untuk keperluan akreditasi.
2. Rancangan Data Warehouse dalam bentuk Information Package dan Skema Data Mart telah siap untuk diimplementasikan untuk membangun Data Warehouse dan aplikasi Intelegensi Bisnis yang nantinya dapat digunakan untuk membantu Manajemen dalam pengambilan keputusan strategis Universitas Atma Jaya Yogyakarta.

DAFTAR PUSTAKA

- Asif, Saadia, 2010, *An Overview Of Business Intelligence*, <http://www.inforica.com/in/download/bipresentation.pdf>. diakses pada tanggal 10 Maret 2010.
- BAN-PT, *Borang dan Instrumen Terbaru*, ban-pt.depdiknas.go.id/index.php?option=com_content&view=article&id=57&Itemid=63&lang=in. diakses pada tanggal 4 Maret 2010.
- Inmon, W.H, 1996. *Building the Data warehouse Second Edition*, John Wiley & Sons, Inc, New York.
- Irmansyah, Farid, 2003. Pengantar *Database*, <http://ilmukomputer.com/umum/faried-database.php> diakses pada tanggal 16 September 2005
- Psomas, Nikos, 2002. *Oracle9iAS: Discoverer for End Users*, Hamsa Venkatachalam, USA.
- Ponniah, Paulraj, 2001. *Data Warehousing Fundamentals*, John Wiley & Sons, Inc, New York.
- Power, D.J., 2002, *A Brief History of Decision Support Systems*, <http://www.dssresources.com/history/dsshhistory.html>. diakses pada tanggal 11 Maret 2010.
- Tim Portofolio Akreditasi Institusi Universitas Atma Jaya Yogyakarta, 2007, *Portofolio Akreditasi Institusi Universitas Atma Jaya Yogyakarta*, tidak diterbitkan.
- Tim SIA Universitas Atma Jaya Yogyakarta, 2007, *Laporan Analisis Kebutuhan Sistem Informasi Akademik Universitas Atma Jaya Yogyakarta*, tidak diterbitkan.